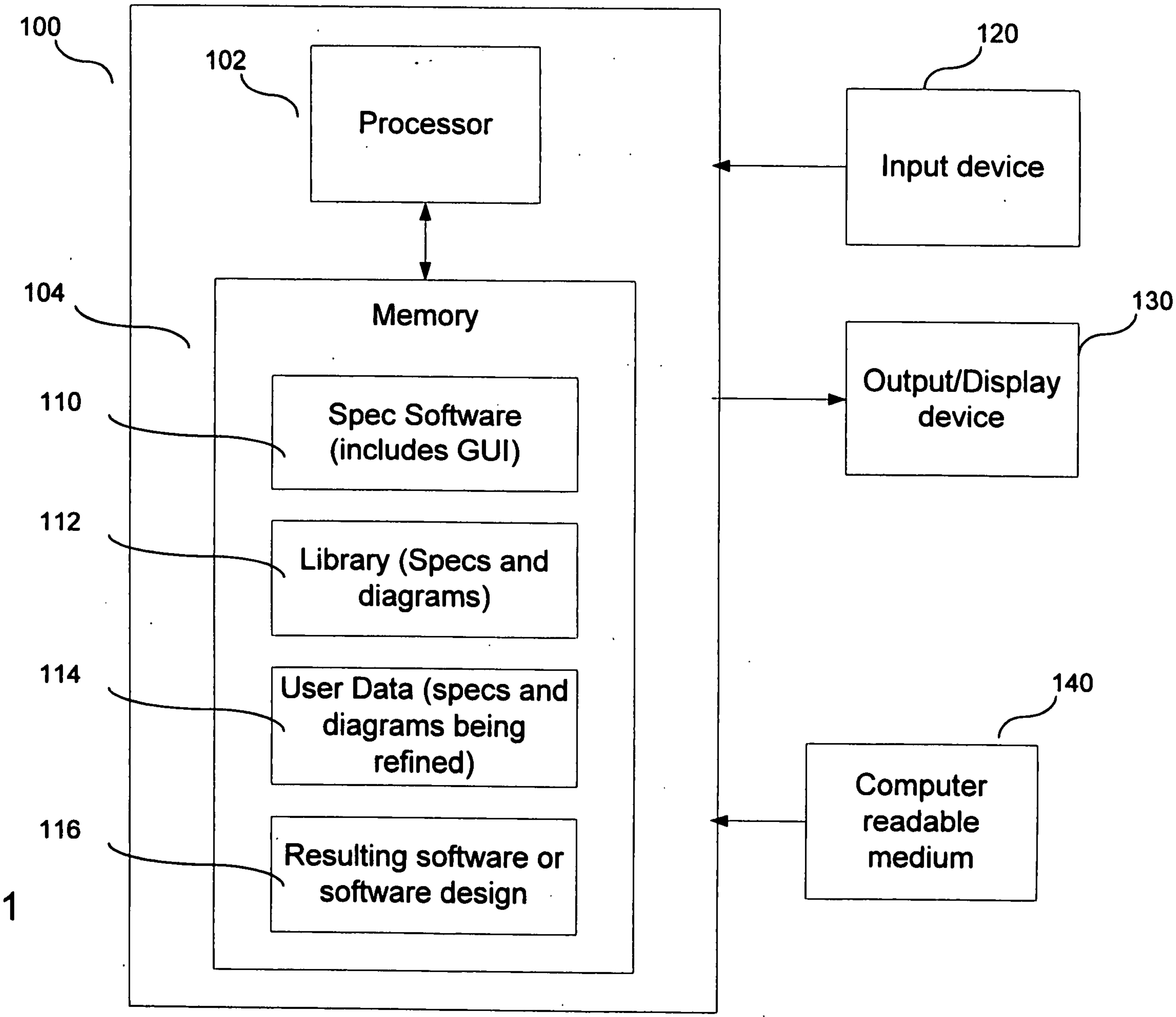

100
102
Processor
104
Memory
110
Spec Software (includes GUI)
112
Library (Specs and diagrams)
114
User Data (specs and diagrams being refined)
116
Resulting software or software design
120
Input device
130
Output/Display device
140
Computer readable medium

Fig. 1

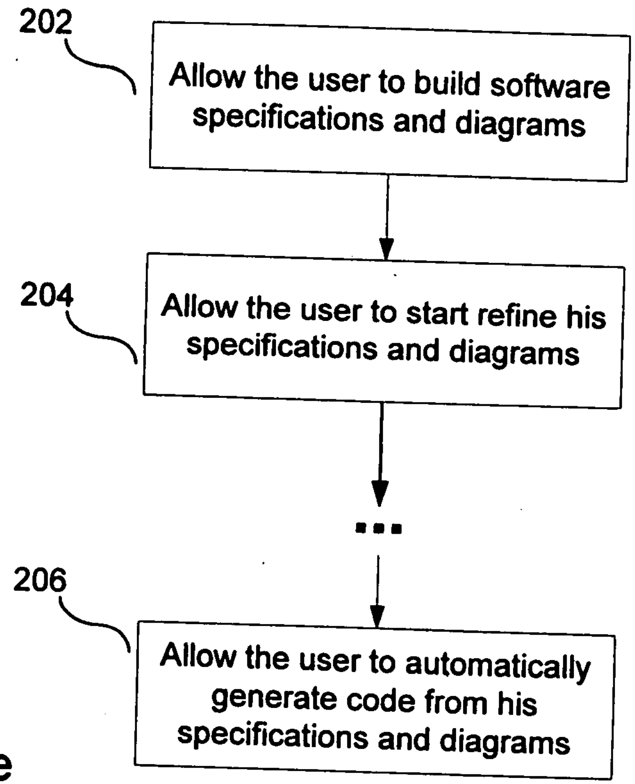


Fig. 2(a)
Spec Software

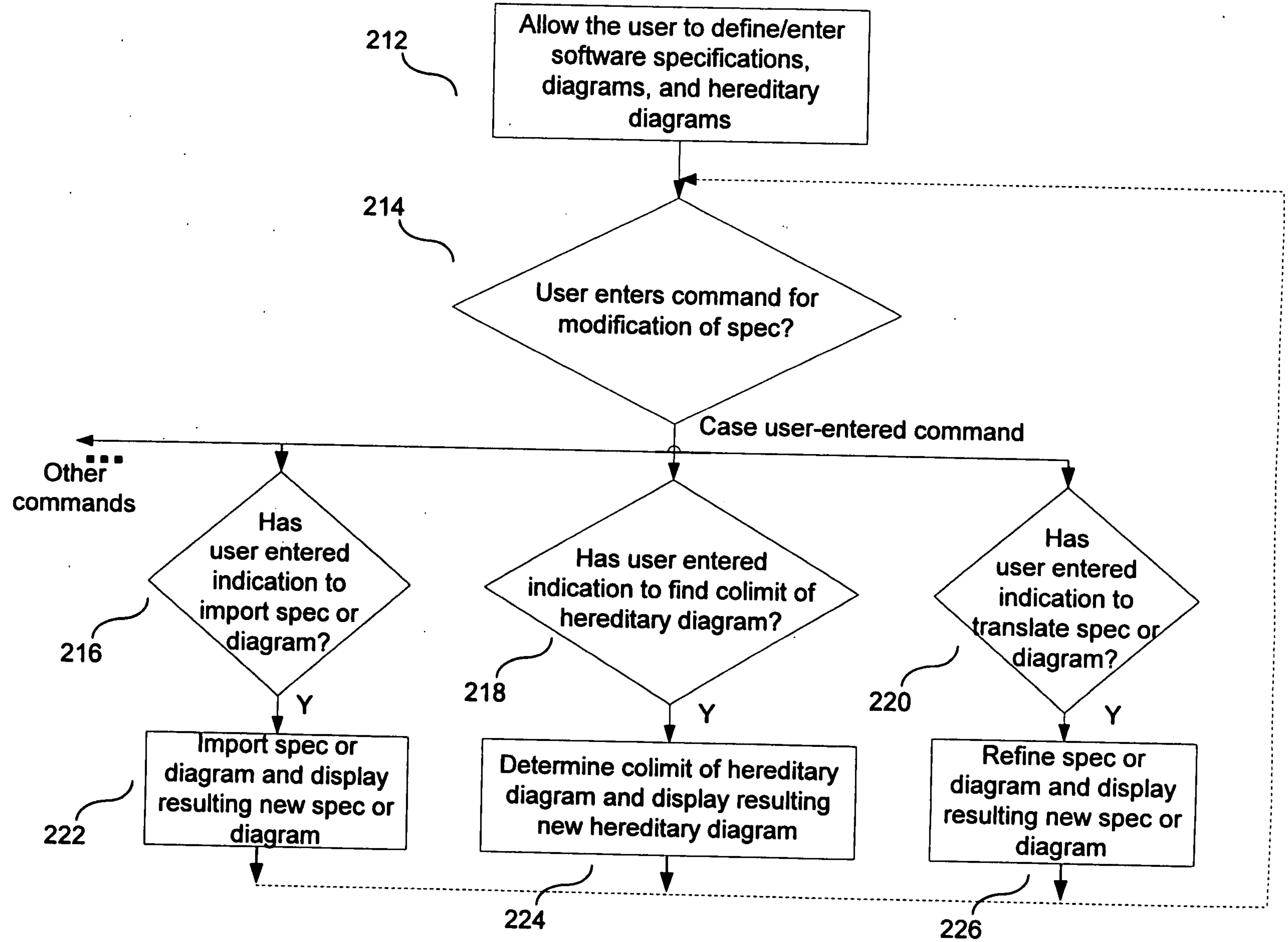
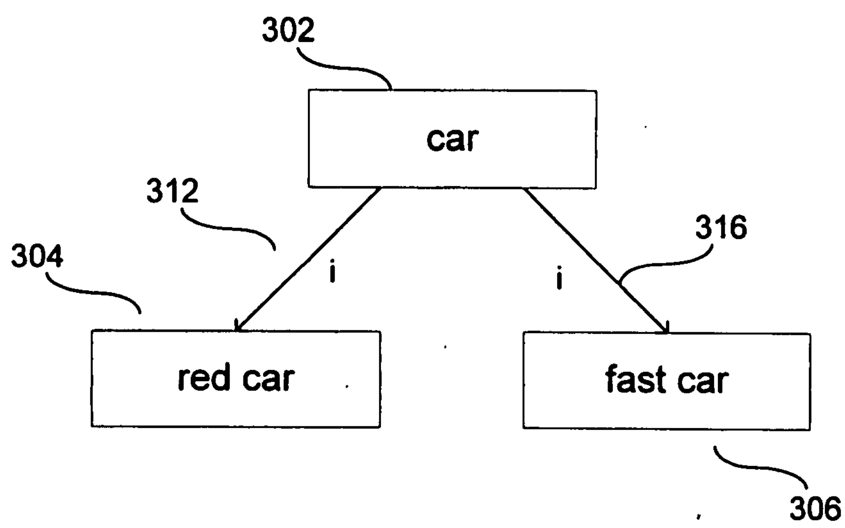
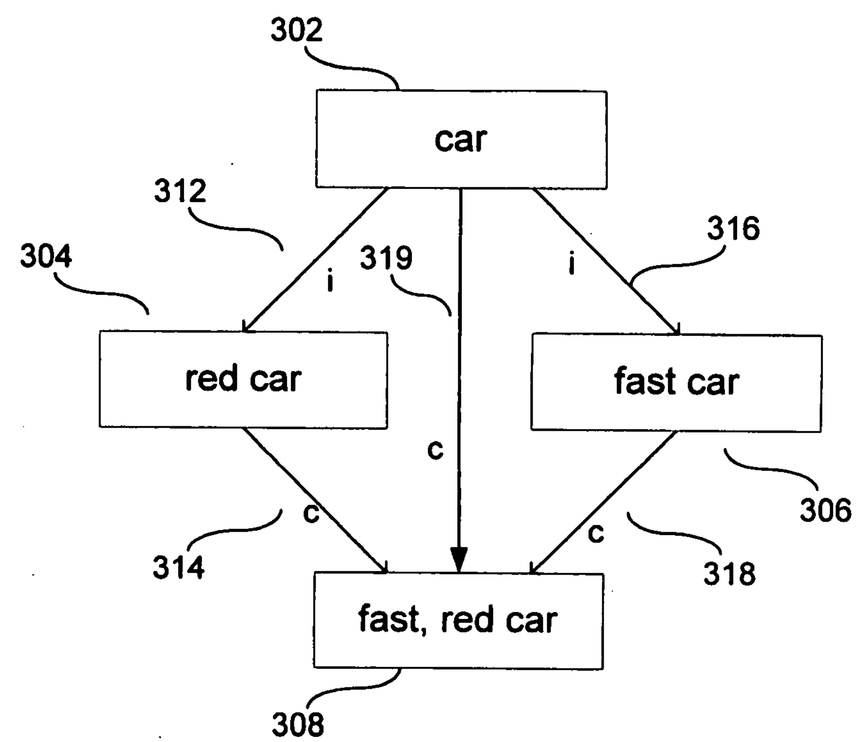


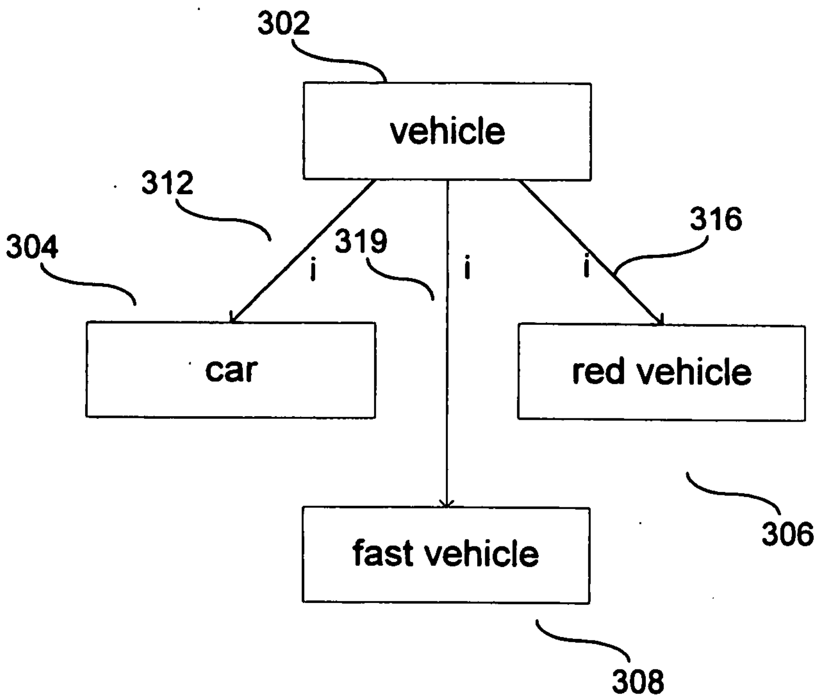
Fig. 2(b)
Spec Software



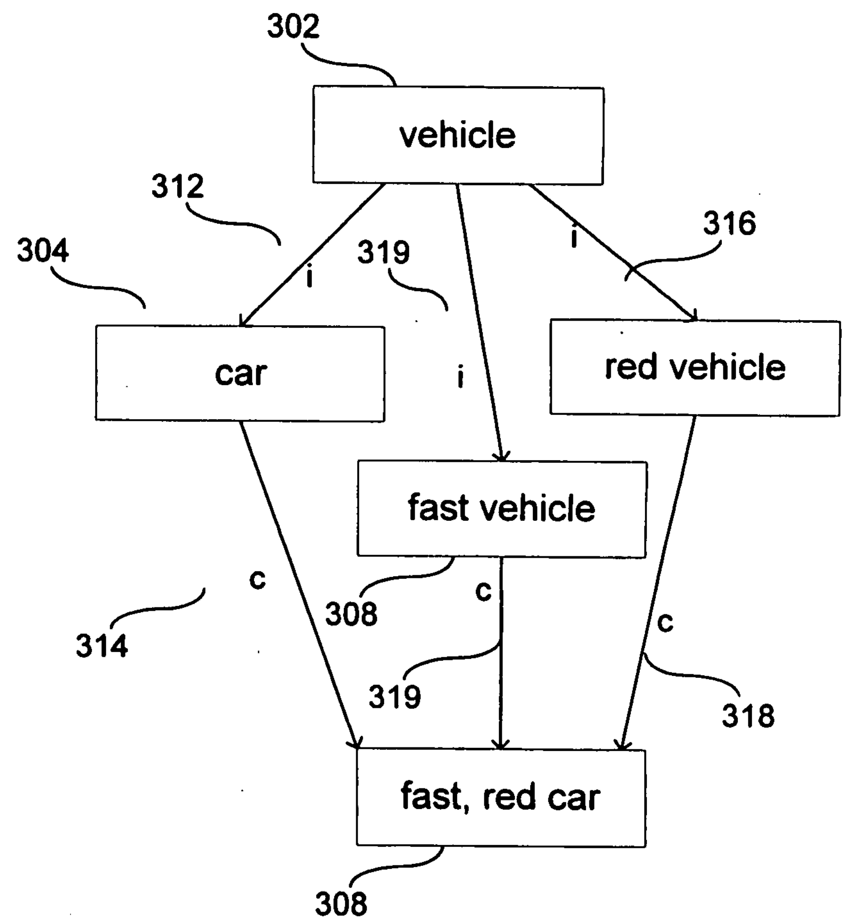
Refining a Specification
Fig. 3(a)



Example of Using a Colimit to
Combine Refined Specifications
Fig. 3(b)

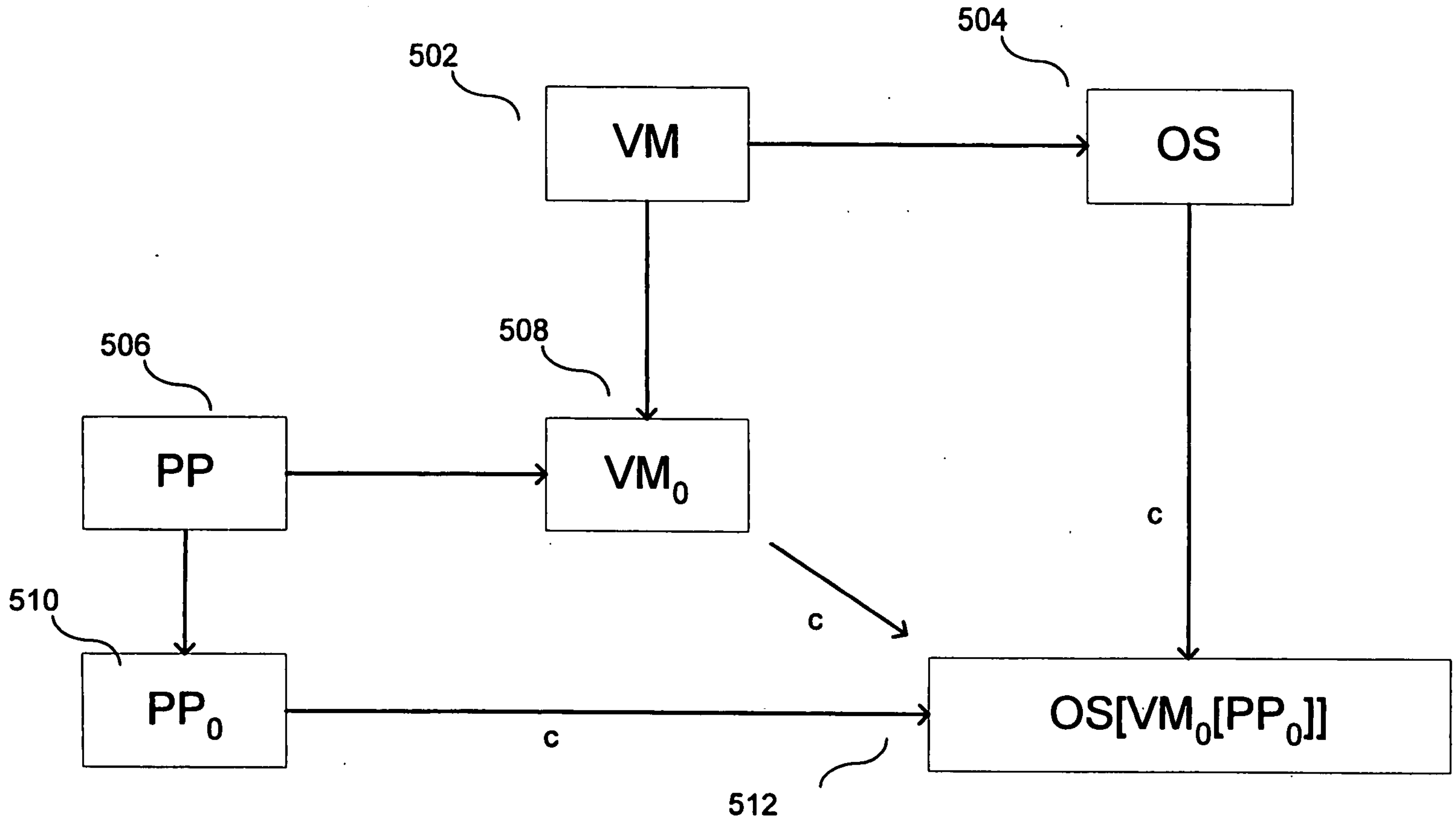


Refining a Specification
Fig. 4(a)

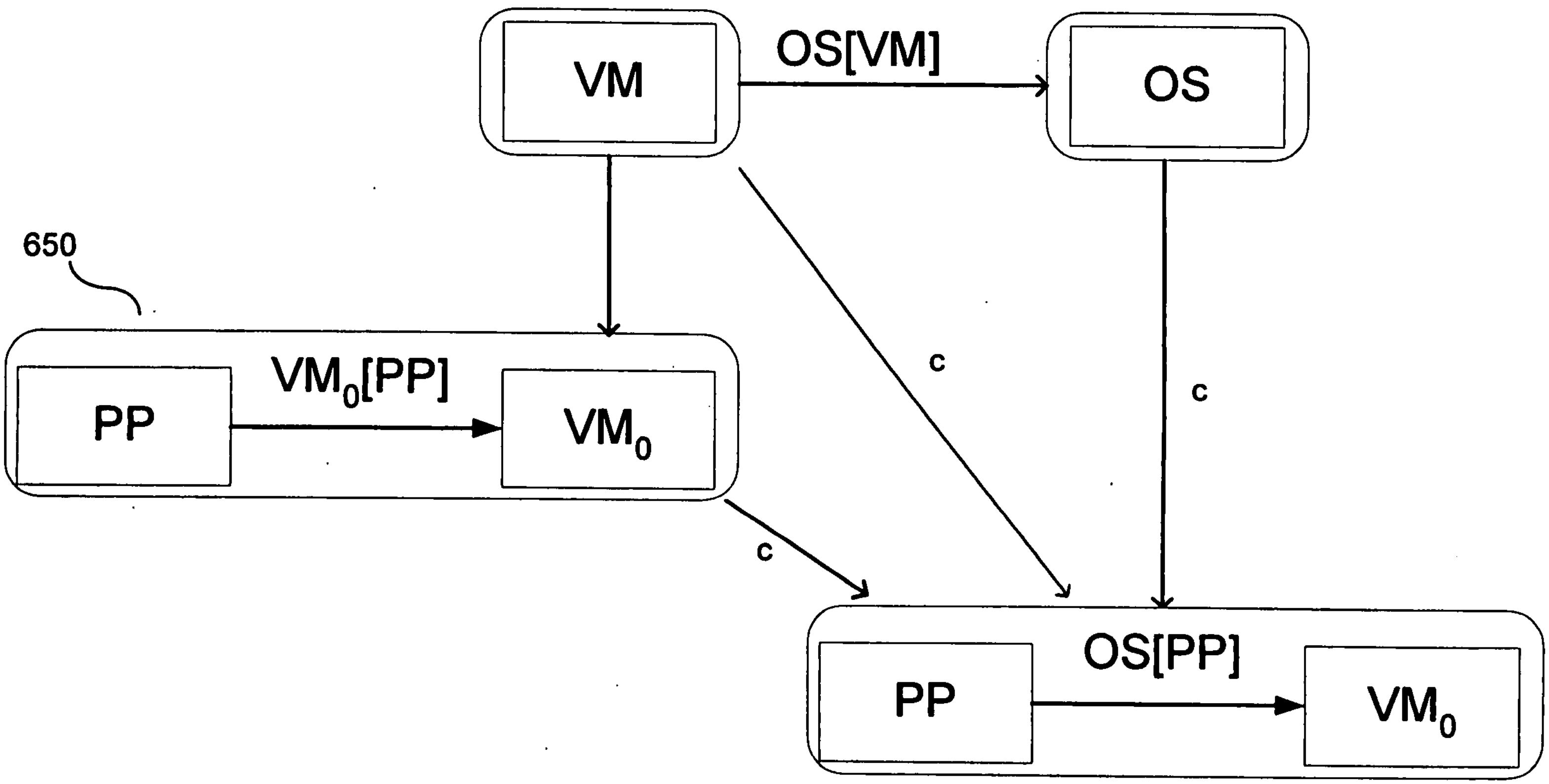


Example of Using a Colimit to
Combine Refined Specifications
Fig. 4(b)

0965172.091900

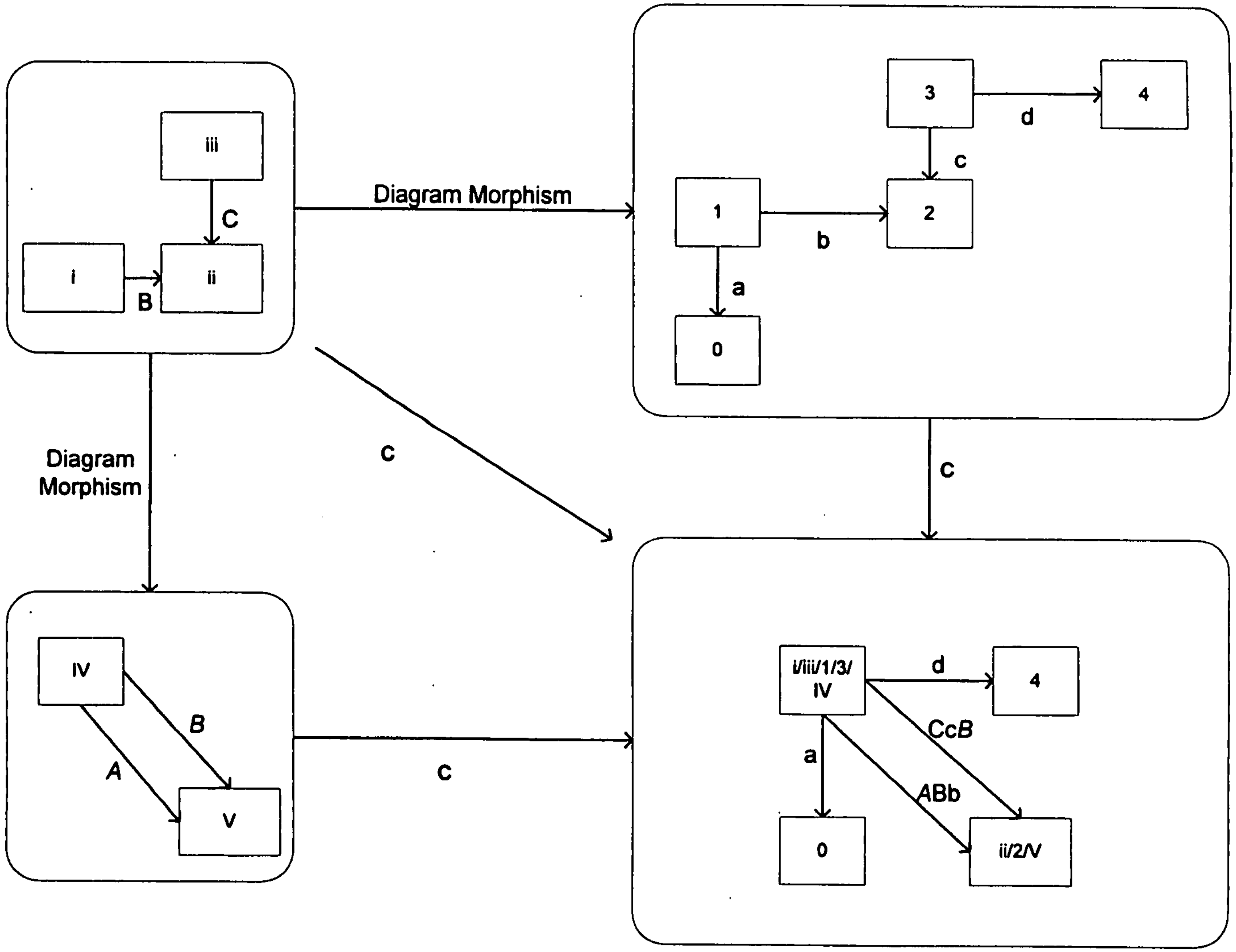


Example Colimit of Specifications
Fig. 5



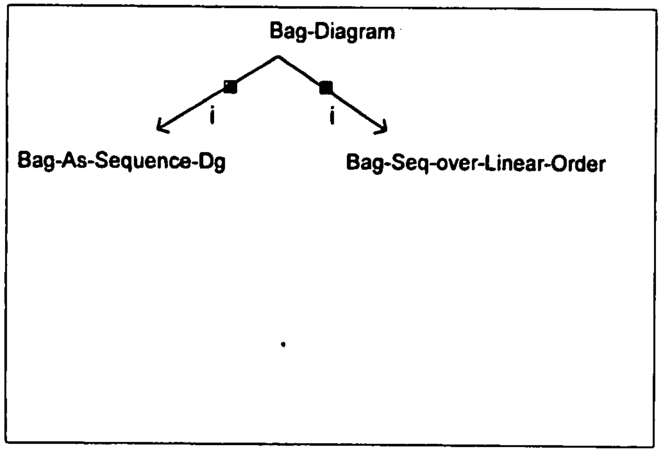
Example Colimit of Diagrams of Diagrams

Fig. 6



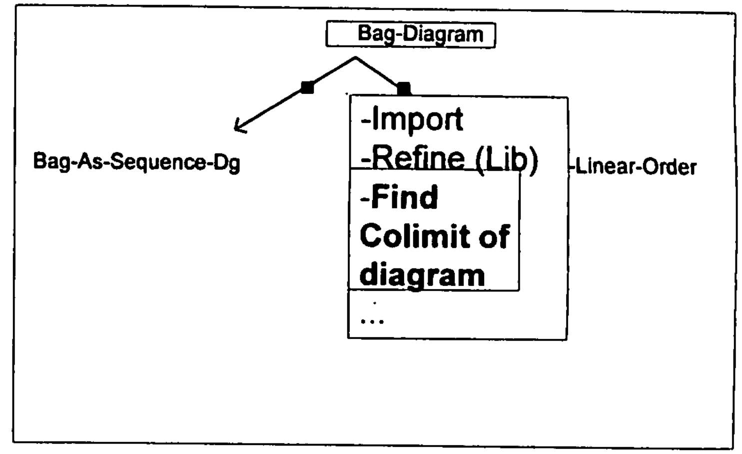
Example of Taking the Colimit of Hereditary Diagrams
Fig. 7

0965179.091900



Example user interface showing a hereditary diagram

Fig. 8(a)

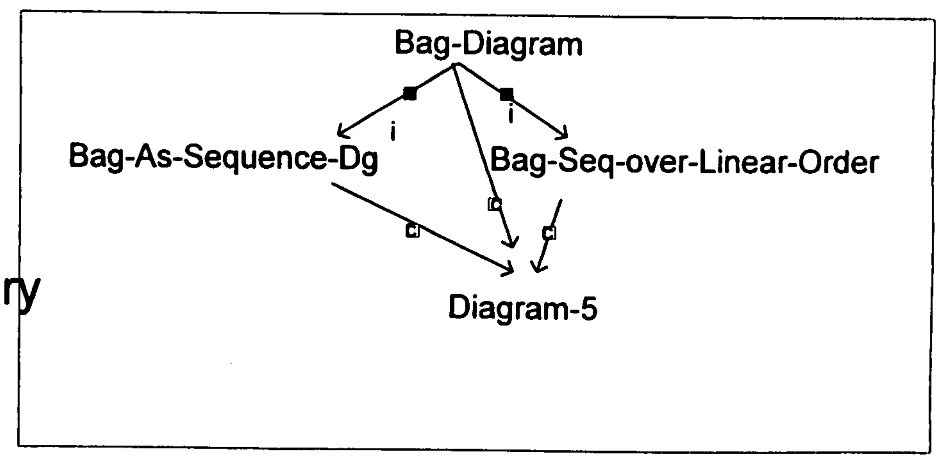


Example user interface showing a hereditary diagram (interface for user to indicate "find colimit" operation)

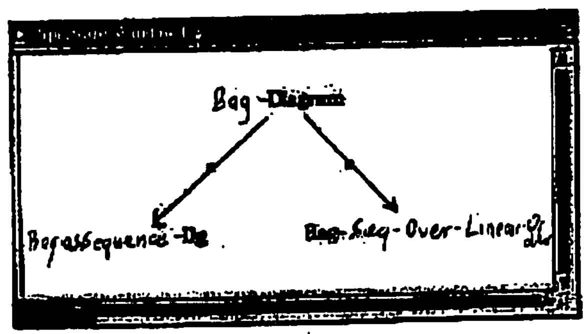
Fig. 8(b)

Example user interface showing a hereditary diagram after the user indicates a "find colimit" operation for the hereditary diagram and the colimit operation is performed

Fig. 8(c)

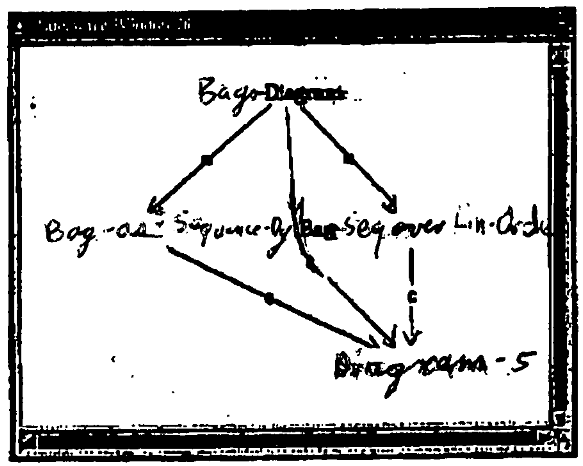


0955172.091900



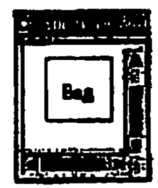
Hereditary diagram

Fig 9(a)



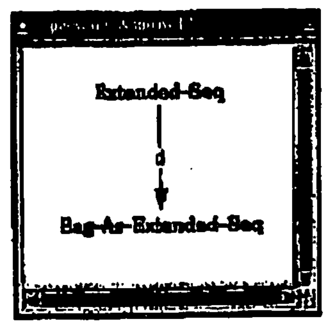
Hereditarydiagram, including colimit

Fig 9(b)



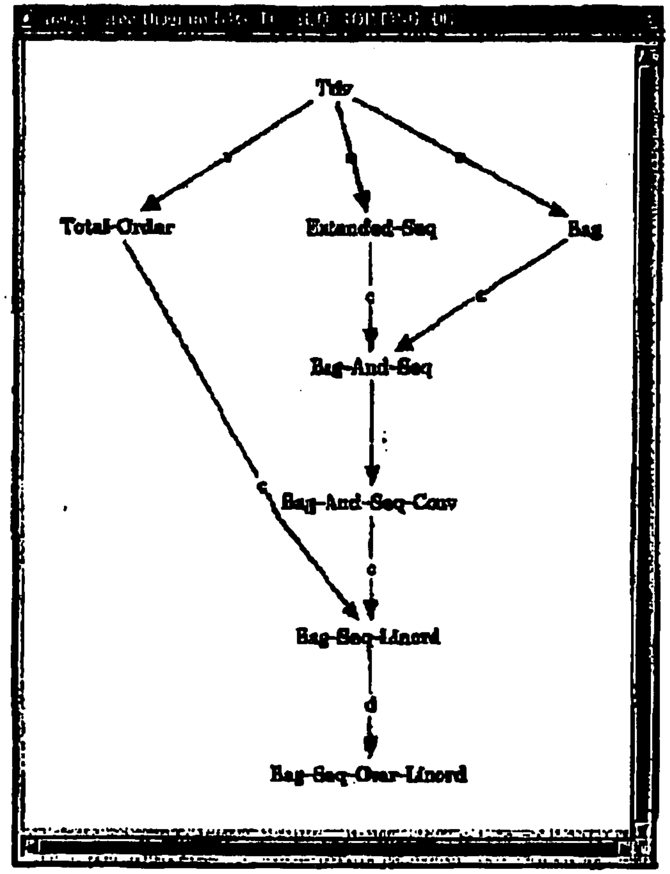
Bag diagram
(obtained by expanding node
Bag-Diagram
in Hereditary diagram)

Fig 9(c)



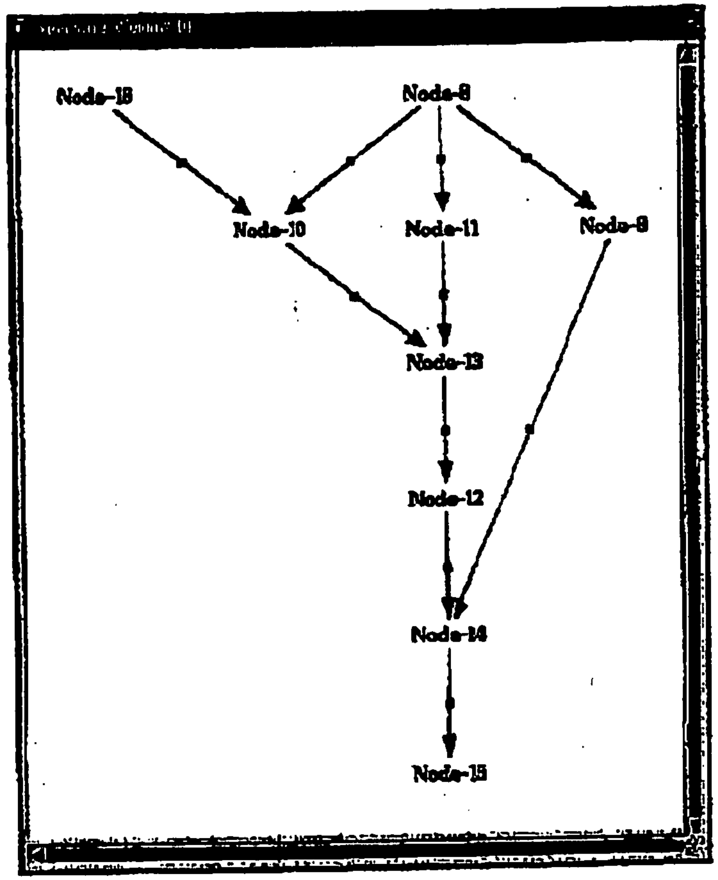
Bag-as-Sequence diagram
(obtained by expanding node
Bag-as-Sequence-diagram
In Hereditary diagram)

Fig 9(d)



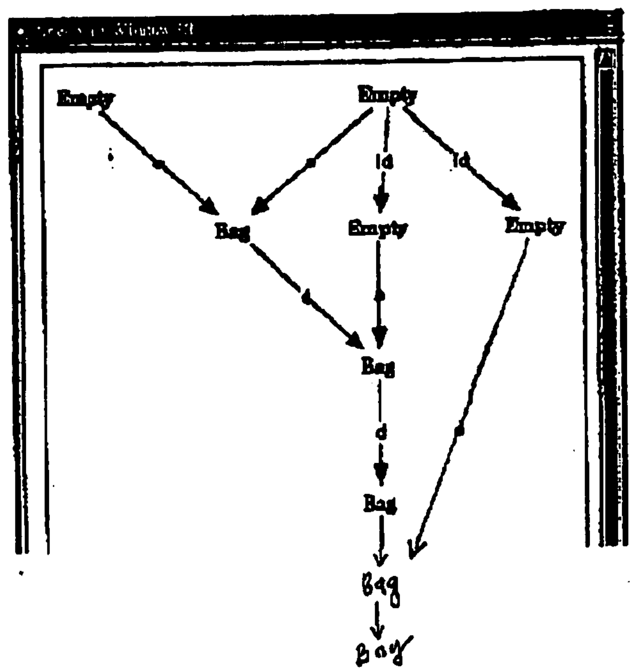
Bag-Seq-over-Linear-Order diagram
(obtained by expanding node
Bag-Seq-over-Linear-Order-diagram
in Hereditary diagram)

Fig 9(e)



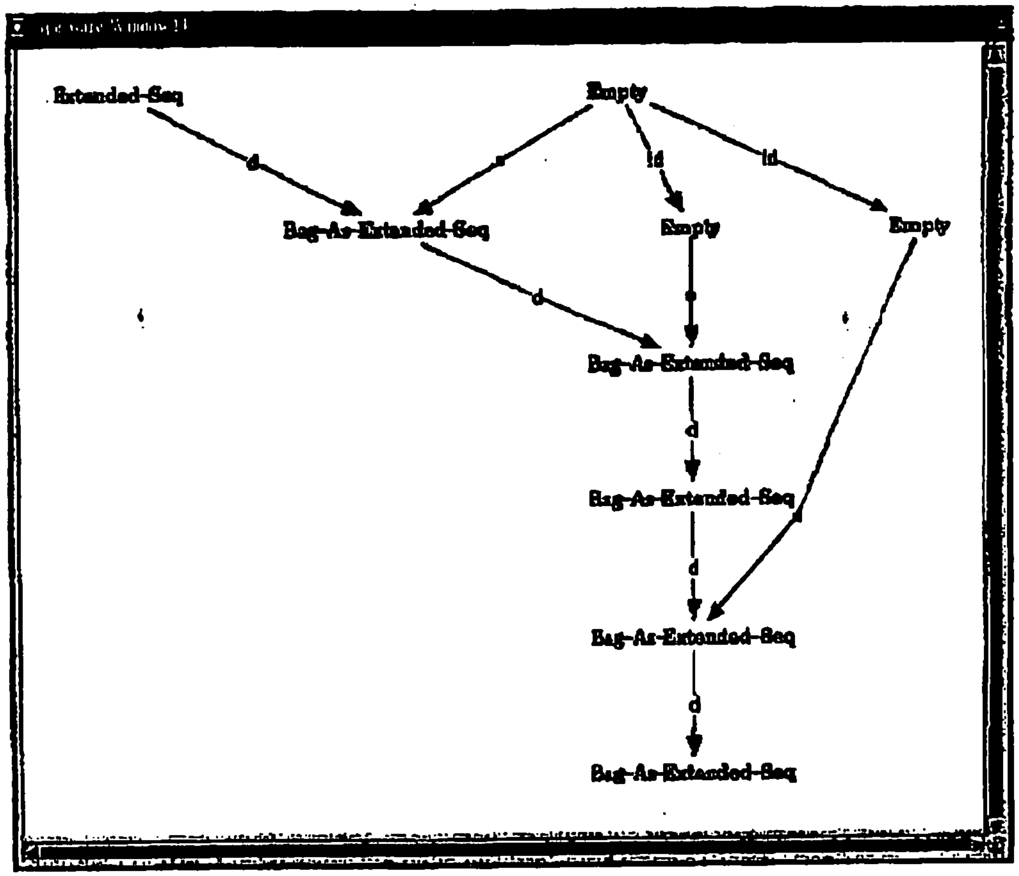
Shape of colimit

Fig 9(f)



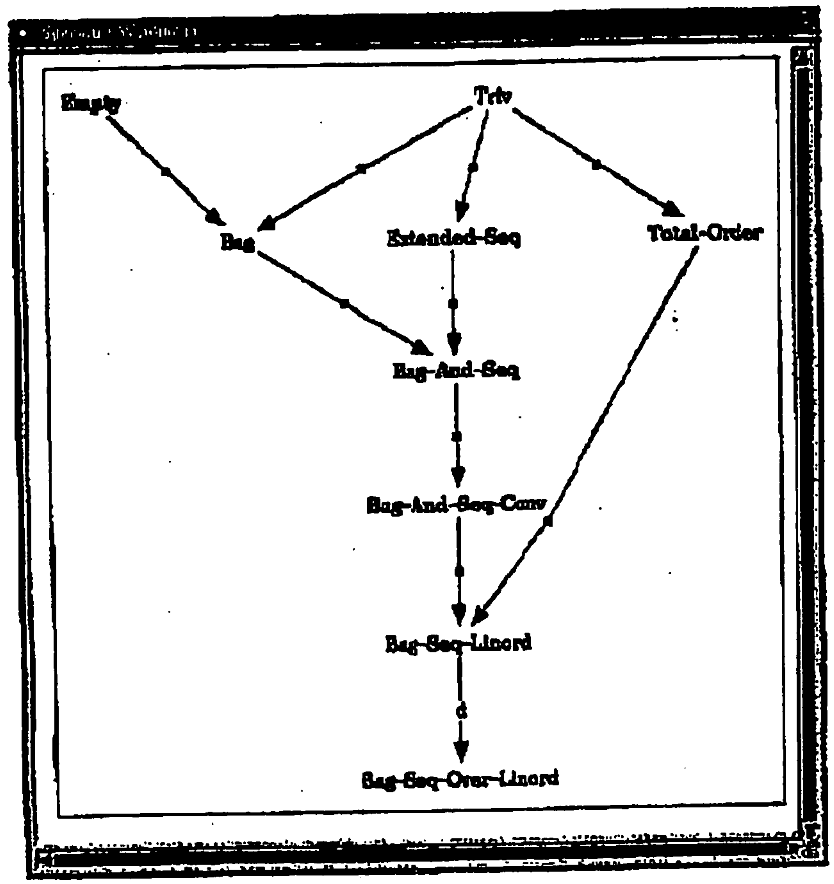
Extended Bag diagram

Fig. 9(g)



Extended Bag-as-Sequence diagram

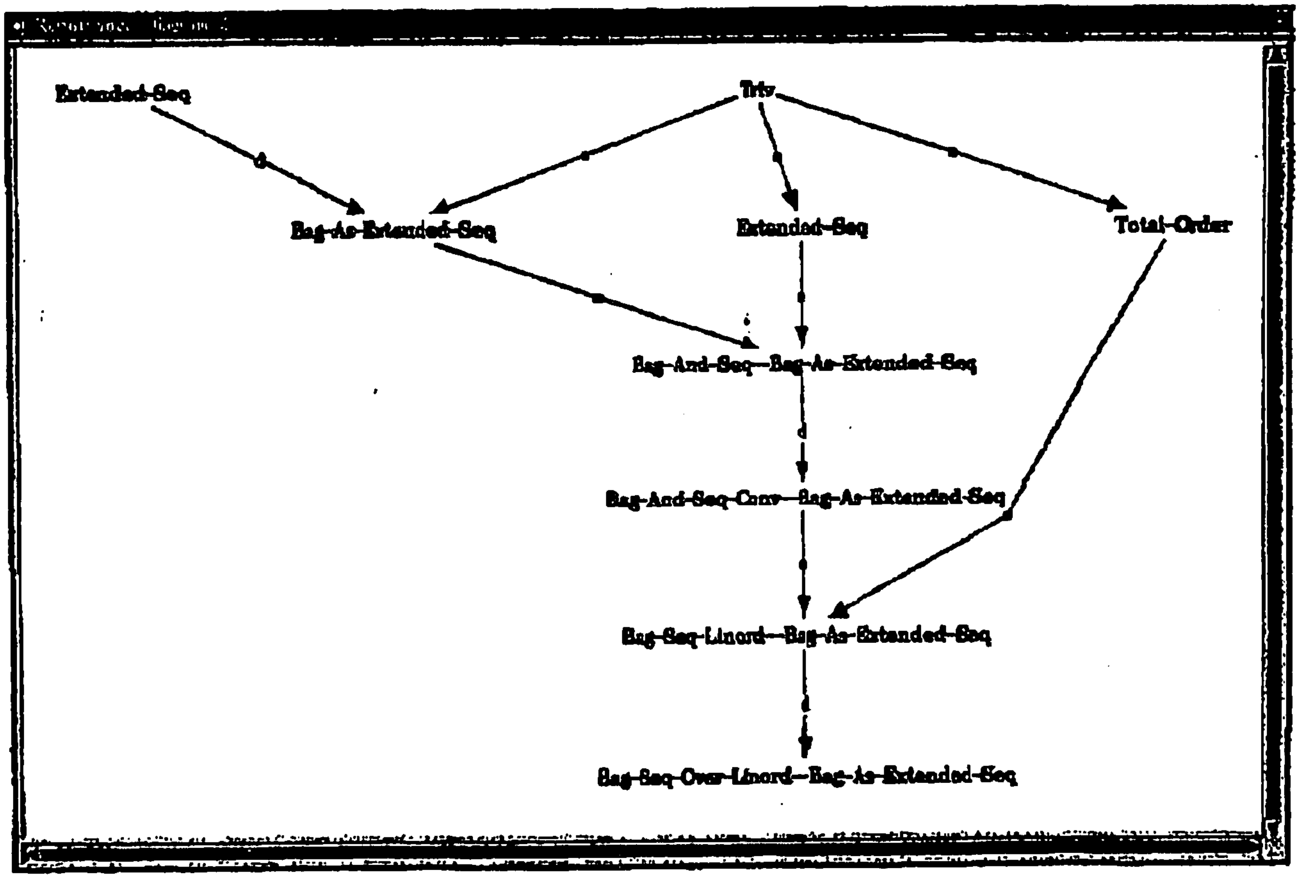
Fig 9(h)



Extended Bag-Seq-over-Linear-Order diagram

Fig 9(i)

09555179.091900



Colimit of Hereditary diagrams
(final result)
Fig 9(j)

1002

User wants to take the colimit of a hereditary diagram (see Fig. 8(b))

1004

I

From the hereditary diagram, extract the shapes of its nodes and the functors between them to yield a diagram of shape categories. (see Fig. 11) Determine the colimit of the diagram of shape categories. The colimit of shape categories has the same shape as the colimit of diagrams that we are trying to determine (see Fig. 16)

1006

II

For each diagram of the hereditary diagram, determine its extension in accordance with the colimit shape obtained in the previous step. (This brings each of the diagrams to the same shape) (see Fig. 19)

1008

III

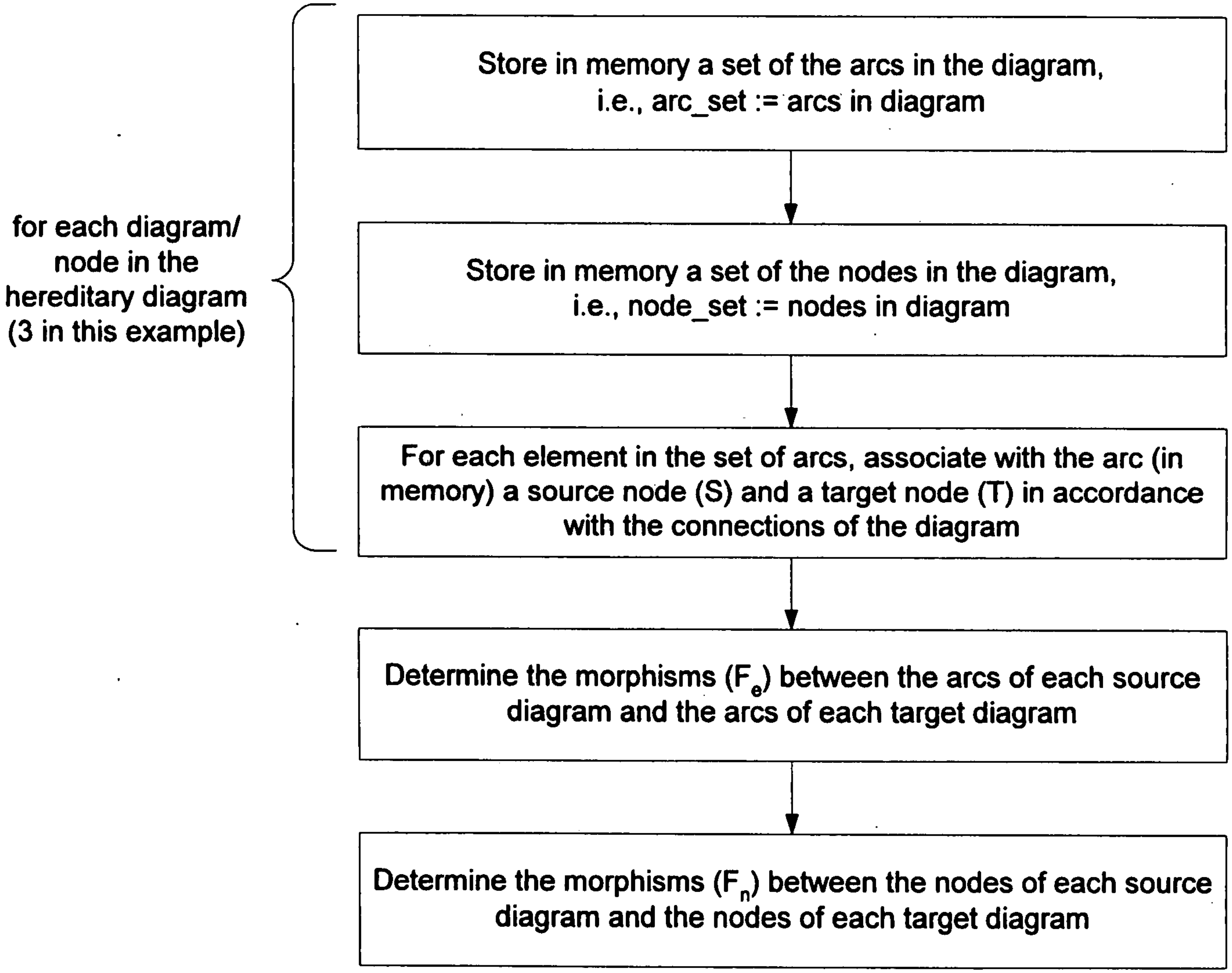
Take pointwise colimit of the extended diagrams to yield the desired colimit of the hereditary diagram. The morphisms of the colimit are uniquely determined by the universality of the pointwise colimits (Can take the pointwise colimit because the extended diagrams are all of the same shape) (See Fig. 22)

1010

Store newly determined colimit in a memory. Add name of newly determined colimit to display (User can view details of colimit if he wishes) (see Fig. 9)

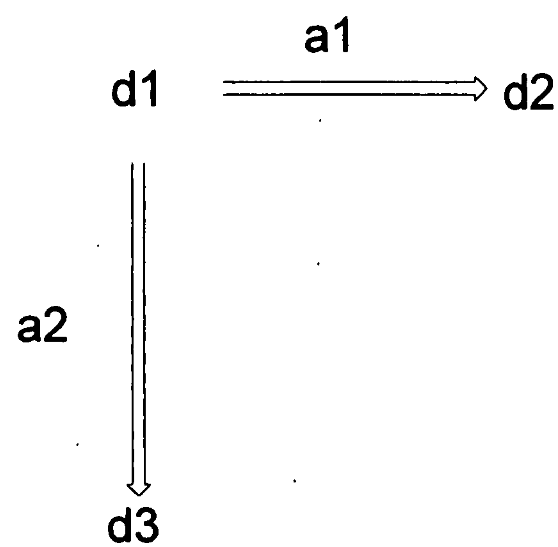
Finding a Colimit of Hereditary Diagrams

Fig. 10



PART I: Extract the shapes and shape functors to yield a diagram of shape categories

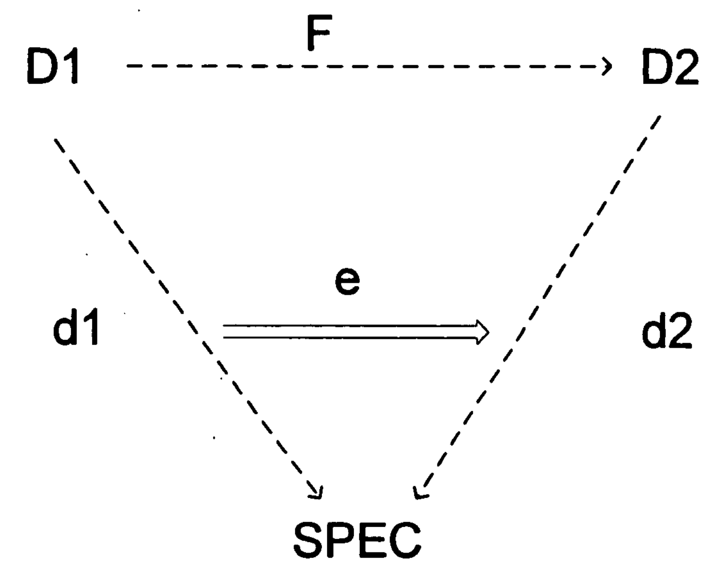
Fig. 11



A Hereditary Diagram: Three Diagrams and Two Arcs.

Each arc a1 and a2 represents a shape morphism having 1) a shape functor (such as F) and 2) a natural shape transformation (such as e: d1 --> d2)

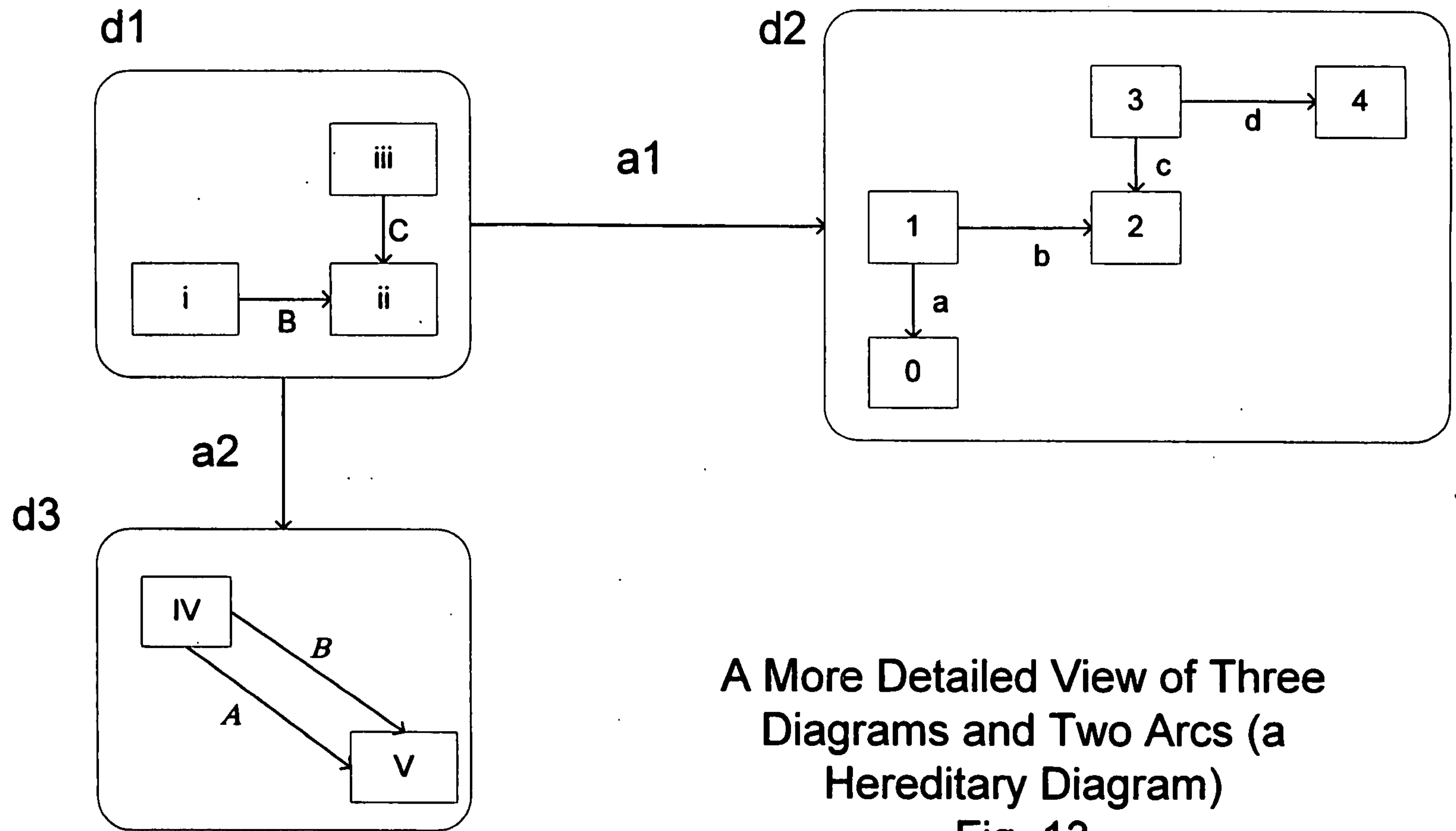
Fig. 12(a)



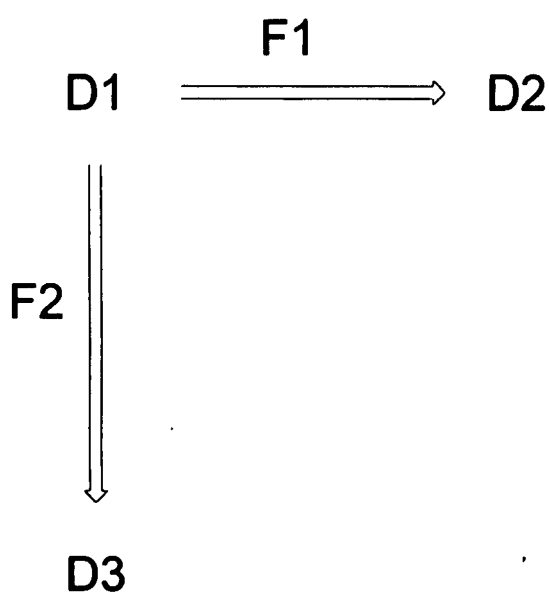
A Shape Morphism

where d1 and d2 are diagrams,
F is a shape functor,
e is a natural transformation from d1 to (d2 composed with F)
D1 and D2 are shape categories of diagrams, and SPEC is the category Spec

Fig. 12(b)

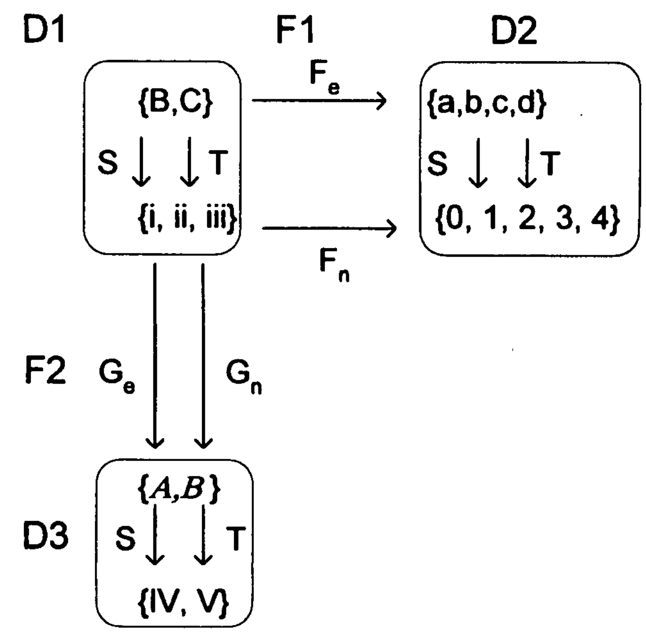


A More Detailed View of Three Diagrams and Two Arcs (a Hereditary Diagram)
Fig. 13



Extract the Shapes and Shape Functors (D1 is shape of diagram d1, F1 is shape functor)

Fig. 14



More Detailed View of Extracting the Shapes and Shape Functors (continued on Figs. 15(b)-15(d))

Fig. 15(a)

Arcs: B -> b
C -> c
Nodes: i -> 1
ii -> 2
iii -> 3

Mapping for F1
Fig. 15(b)

Arcs: B -> *A*
C -> *B*
Nodes: i -> IV
ii -> V
iii -> IV

Mapping for F2
Fig. 15(c)

Arc	B	C	a	b	c	d	*A*	*B*
Source	i	iii	1	1	3	3	IV	IV
Target	ii	ii	0	2	2	4	V	V

Source (S) and Target (T) Functions for Hereditary Diagrams
Fig. 15(d)

Compute colimit of sets of nodes: Store in memory a disjoint union of all nodes (ignore arcs). Determine the equivalence relations identifying those nodes that are connected by some arc of the hereditary diagram. All nodes of the diagrams that fall in the same equivalence class are identified as a single node in the colimit

↓

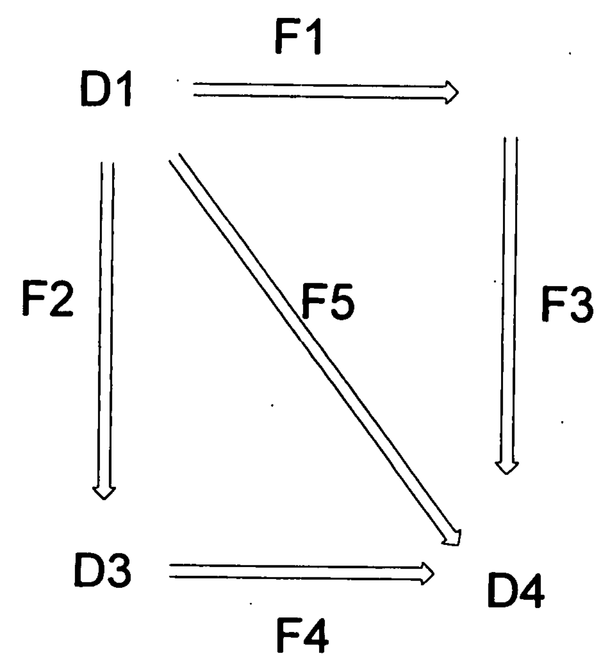
Compute colimit of sets of arcs: Store in memory a disjoint union of all arcs (ignore nodes). Determine the equivalence relations identifying those arcs that are connected by some arc of the hereditary diagram. All arcs of the diagrams that fall in the same equivalence class are identified as a single arc in the colimit

↓

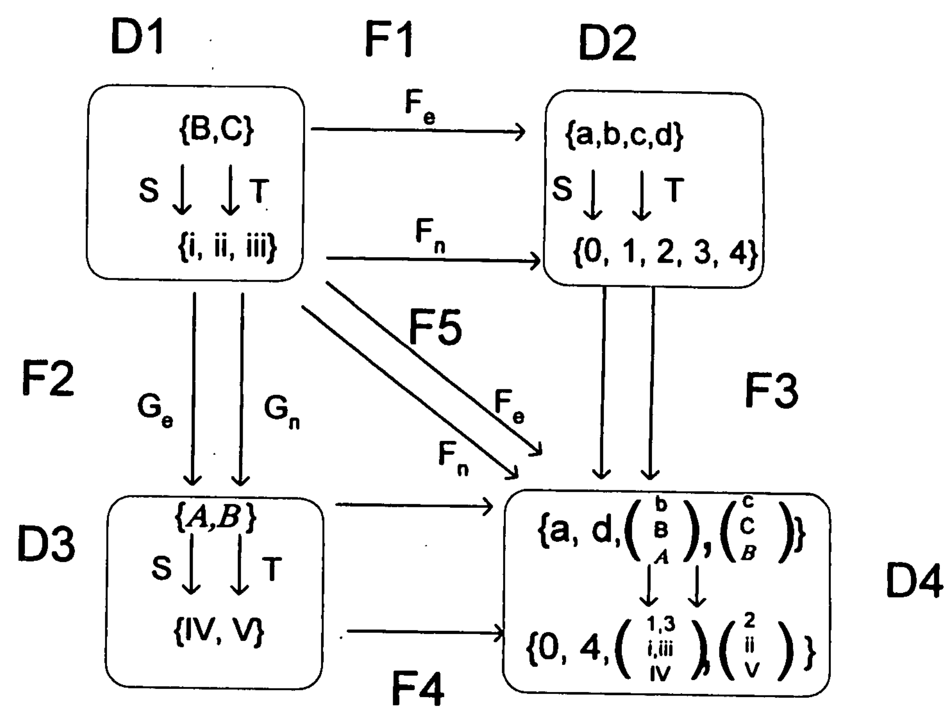
Consider the relationship between the equivalence classes of arcs and nodes. For each arc in the colimit, the universal property of the colimit (of the sets of arcs) determines a source node (S) and a target node (T)

PART I: Determine the colimit of the diagram of shape categories.

Fig. 16



More Detailed View of Taking the Colimit
Fig. 17



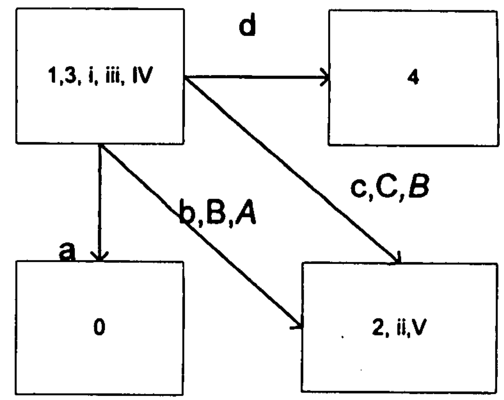
More Detailed View of Taking the Colimit
(See Figs 18(b)-(f))

Fig. 18(a)

0965517?.091900

Arc	a	d	b A *A*	c C *B*
Source	1,3 i,iii IV	1,3 i,iii IV	1,3 i,iii IV	1,3 i,iii IV
Target	0	4	2 ii V	2 ii V

Source (S) and Target (T) Functions for Shape Colimit D4
Fig. 18(b)



The Colimit D4 of the Shape Diagrams
Fig. 18(c)

Arcs: a -> a
d -> d
b -> b,B,*A*
c -> c,C,*B*

Nodes: 0 -> 0
1 -> 1,3, i,iii, IV
2 -> 2, ii,V
3 -> 1,3, i,iii, IV
4 -> 4

Mapping for F3
Fig. 18(d)

Arcs: *A* -> b,B,*A*
B -> c,C,*B*

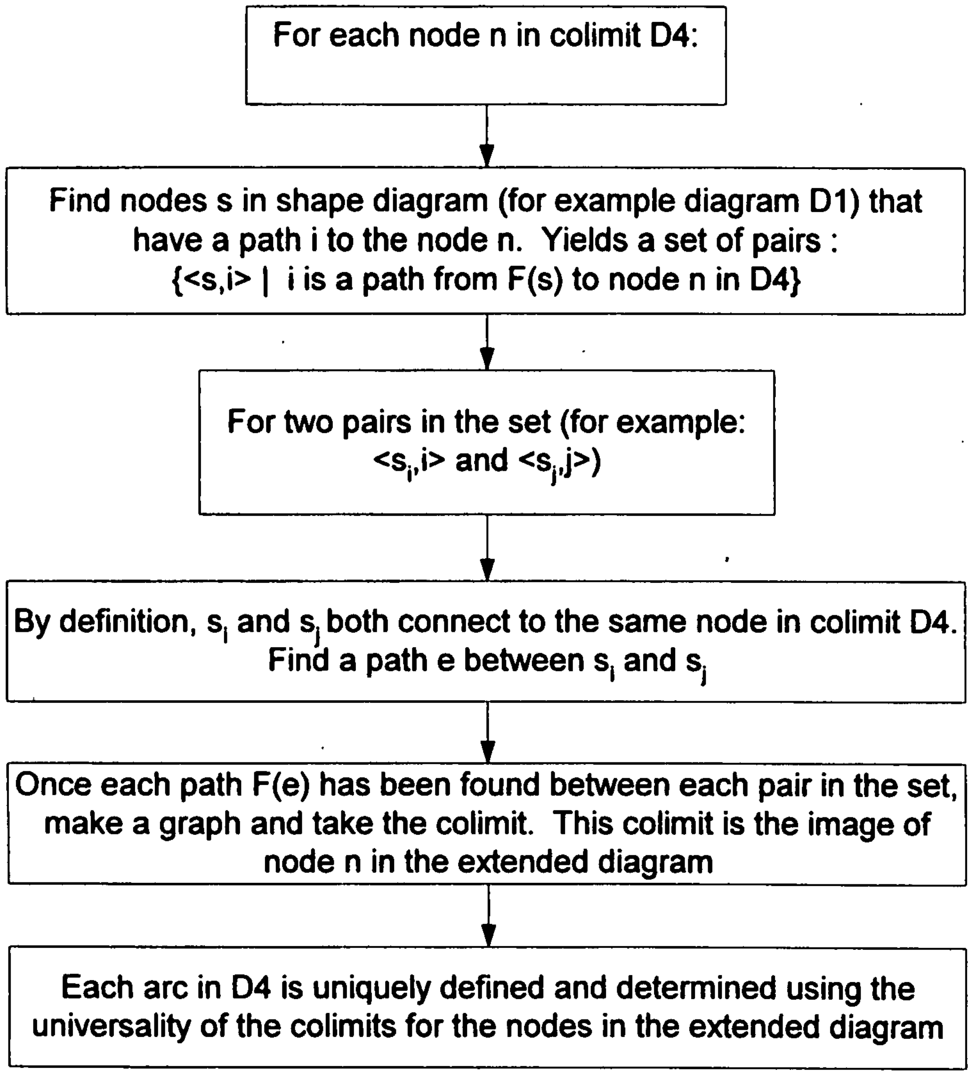
Nodes: IV -> 1,3, i, iii, IV
V -> 2, ii,V

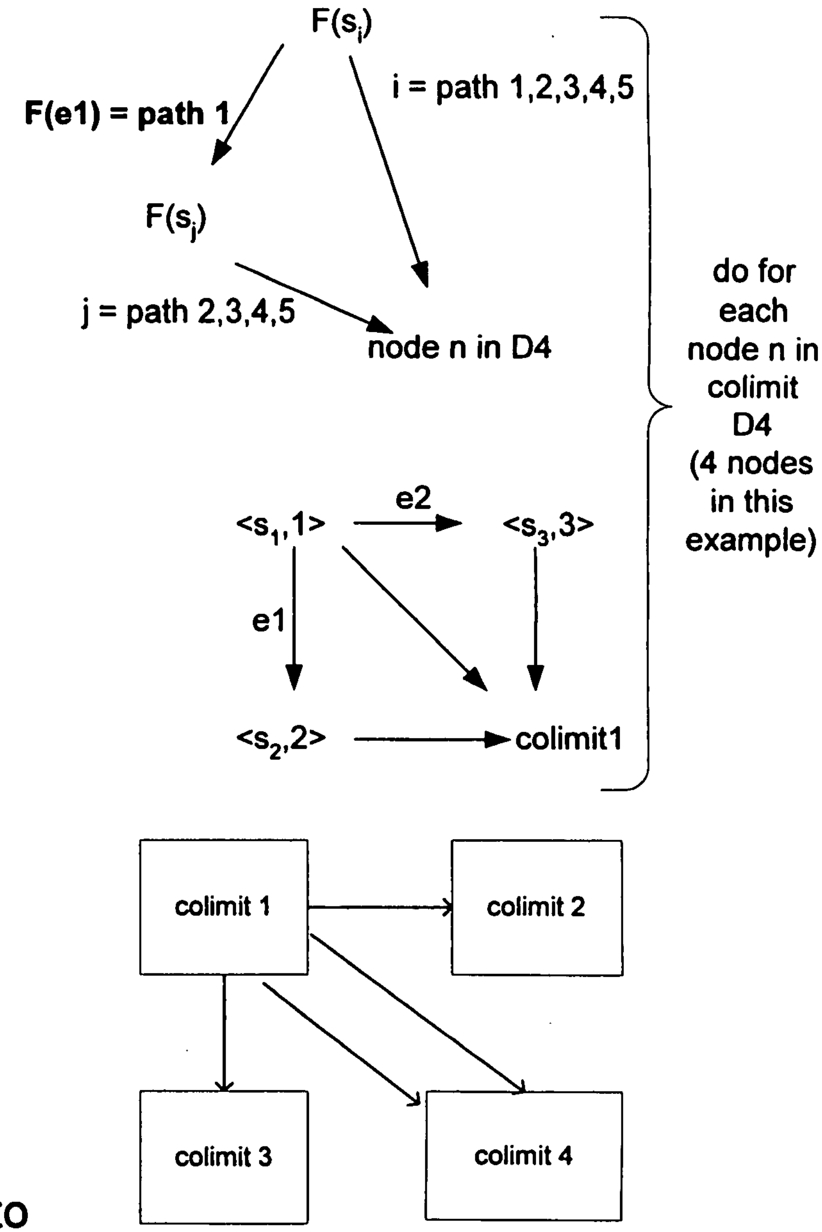
Mapping for F4
Fig. 18(e)

Arcs: B -> b,B,*A*
C -> c,C,*B*

Nodes: i -> 1,3, i, iii, IV
ii -> 2, ii,V
iii -> 1,3, i, iii, IV

Mapping for F5
Fig. 18(f)





extended diagram from example

PART II:Extending one Diagram (repeat to extend each diagram)

Fig. 19

Extension for Diagram d1:

$$\begin{tikzcd}
D1 \arrow[rr] \arrow[dr, Rightarrow, "F5"'] & & SPEC \\
& D4 \arrow[ur, Rightarrow, "Lf5d1"'] &
\end{tikzcd}$$

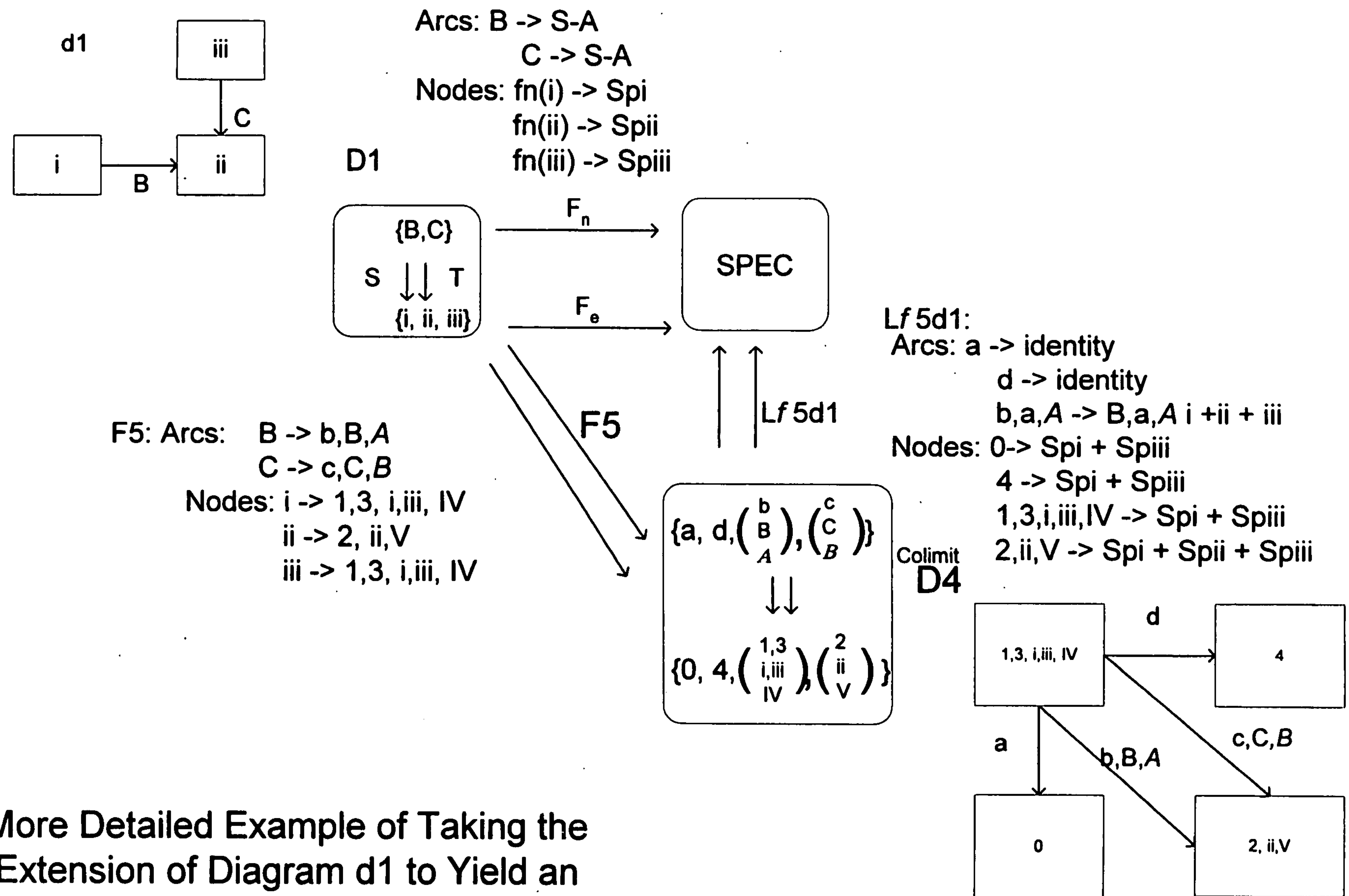
Extension for Diagram d2:

$$\begin{tikzcd}
D2 \arrow[rr] \arrow[dr, Rightarrow, "F3"'] & & SPEC \\
& D4 \arrow[ur, Rightarrow, "Lf3d2"'] &
\end{tikzcd}$$

Extension for Diagram d3:

$$\begin{tikzcd}
D3 \arrow[rr] \arrow[dr, Rightarrow, "F4"'] & & SPEC \\
& D4 \arrow[ur, Rightarrow, "Lf4d3"'] &
\end{tikzcd}$$

Example of Taking the Extension of Each Node of the Hereditary Diagram
Fig 20



More Detailed Example of Taking the Extension of Diagram d1 to Yield an Extended Diagram
Fig. 21(a)

1

For four Nodes in D4

$i \xrightarrow{F_5} \begin{pmatrix} 1,3 \\ i, iii \\ \mathrm{IV} \end{pmatrix}$

$\downarrow a$

Node 0

$\{ \langle i, a \rangle \mid a : F_5(i) = \begin{pmatrix} 1,3 \\ i, iii \\ \mathrm{IV} \end{pmatrix} \xrightarrow{a} 0 \}$

$\langle iii, a \rangle \mid a : F(iii) = \begin{pmatrix} 1,3 \\ i, iii \\ \mathrm{IV} \end{pmatrix} \xrightarrow{a} 0 \}$

$\langle i, a \rangle \rightarrow 0$, $\langle iii, a \rangle \rightarrow 0$, id

$\langle i, a \rangle \rightarrow Sp_i$ + $\langle iii, a \rangle \rightarrow Sp_{iii}$ coproduct (i.e. no arc between th…

Node 4

$\{ \langle i, d \rangle \mid d : F_5(i) = \begin{pmatrix} 1,3 \\ i, iii \\ \mathrm{IV} \end{pmatrix} \xrightarrow{d} 4 \}$

$\langle iii, d \rangle \mid$ —— —— $\}$

same as above

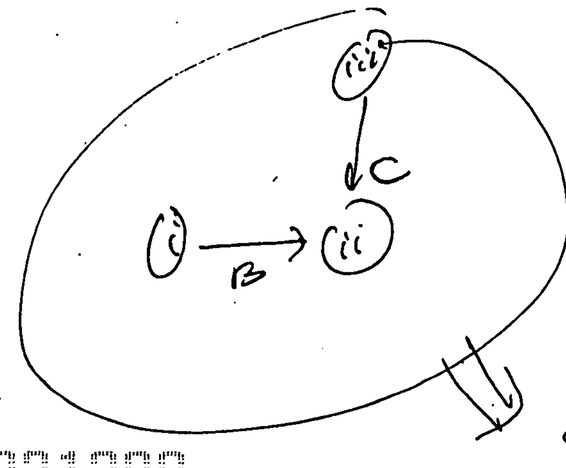
Node $\begin{pmatrix} 1,3 \\ i, iii \\ \mathrm{IV} \end{pmatrix}$

$\{ \langle i, id \rangle, \langle iii, id \rangle \}$

Node $\begin{pmatrix} 2 \\ ii \\ \mathrm{V} \end{pmatrix}$

$\{ (i, BA), (iii, CB) \}$

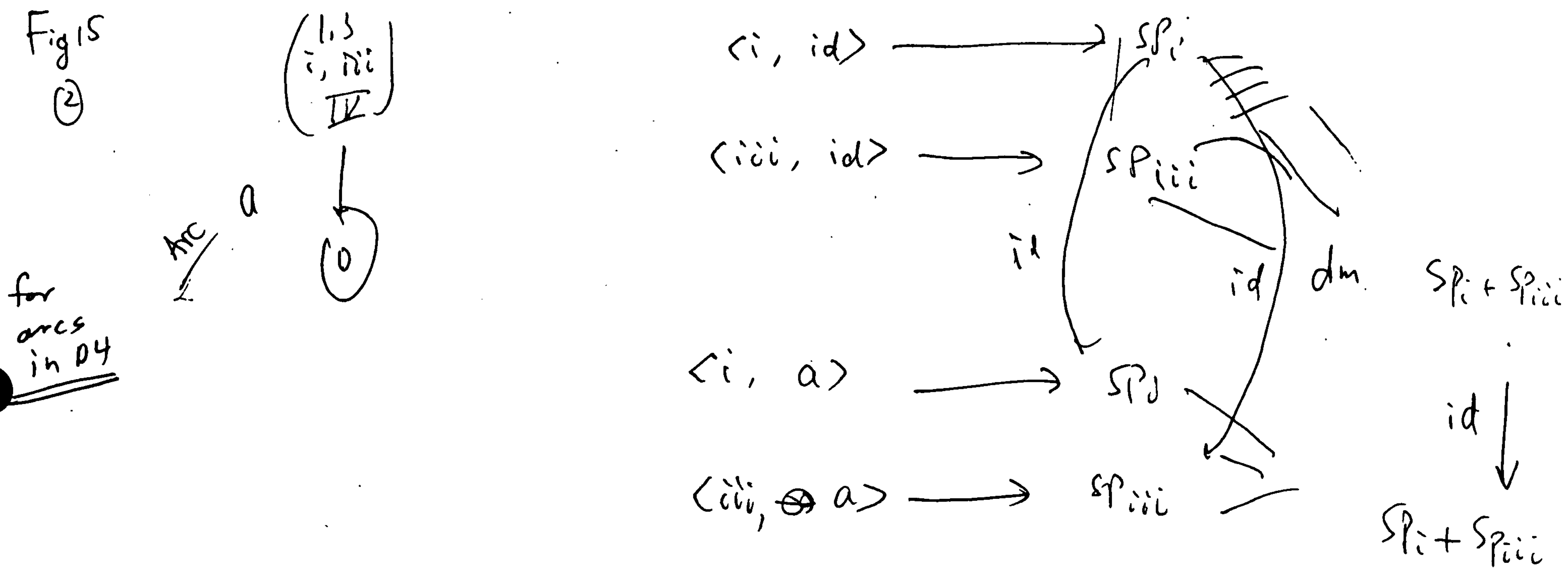
(ii, id)



$Sp_{i+ii+iii}$ colimit

Fig 21(b) Example of Diagram Extension

Fig 15 ②



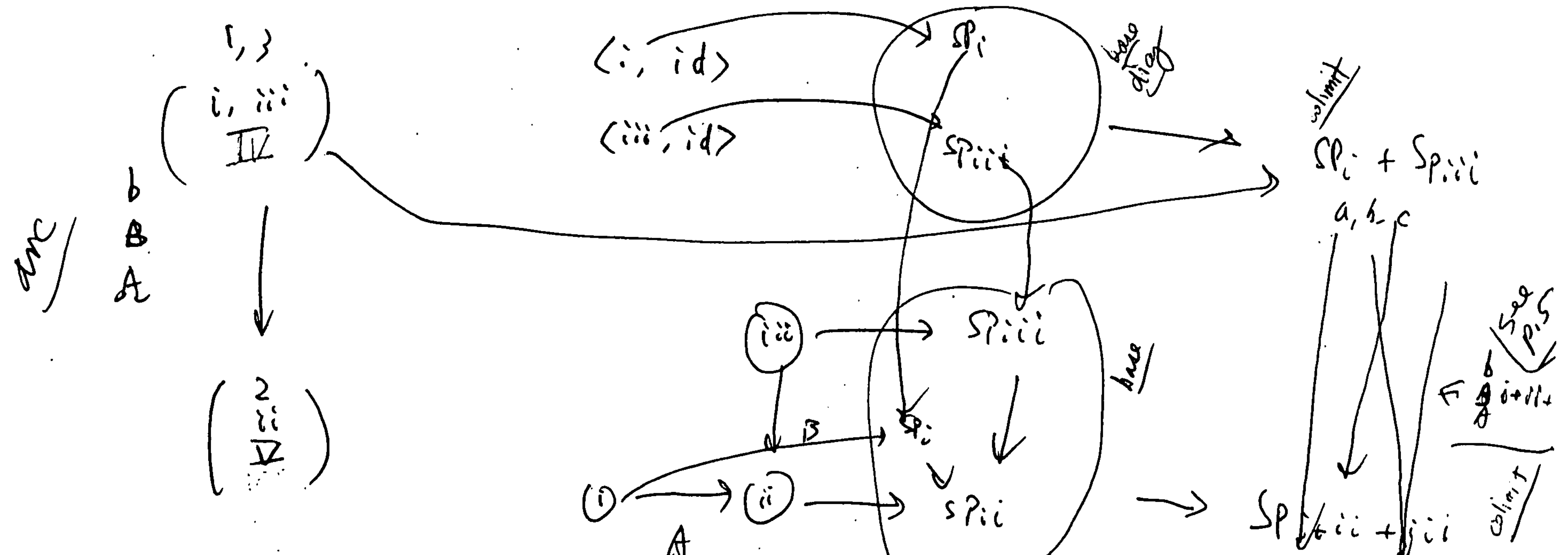
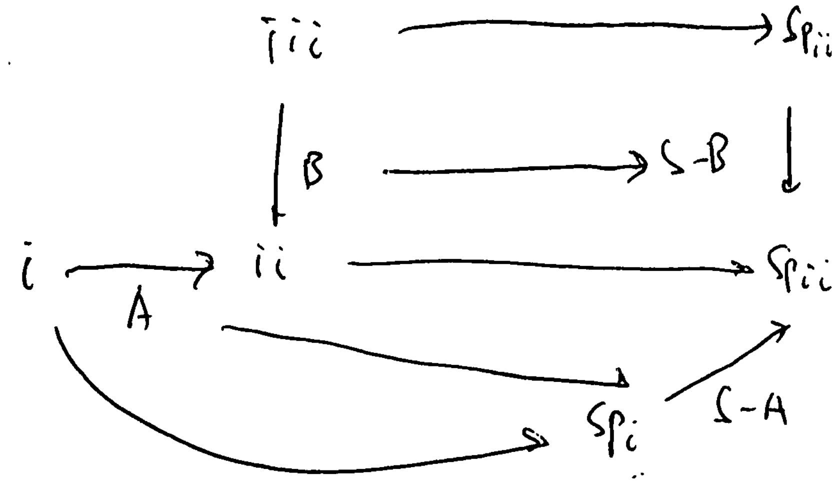


Fig 21(c) Example of Diagram extension (cont)

09655479.091900

$$Sp_i = \left\{ \begin{array}{l} \text{spec } Sp_i \text{ is} \\ \quad \text{sort } S_1 \\ \quad \text{op } f_1 : S_1 \rightarrow \text{boolean} \\ \quad \text{op } f_2 : S_1 \rightarrow \text{boolean} \\ \quad \text{axiom } f_1 \rightarrow f_2 \text{ is} \\ \qquad f_1(x) \Rightarrow f_2(x) \end{array} \right\}$$



Base diagram

$$Sp_{ii} = \left\{ \begin{array}{l} \text{spec } Sp_{ii} \text{ is} \\ \quad \text{sort } S_2 \\ \quad \text{op } g : S_2 \rightarrow \text{boolean} \end{array} \right\}$$

$$Sp_{iii} = \left\{ \begin{array}{l} \text{spec } Sp_{iii} \text{ is} \\ \quad \text{sort } S_1 \\ \quad \text{op } f_1 : S_1 \rightarrow \text{boolean} \end{array} \right\}$$

$$S\text{-}A = \left\{ \begin{array}{lcl} Sp_i & \longrightarrow & Sp_{ii} \\ S_1 & \longrightarrow & S_2 \\ f_1 & \longrightarrow & g \\ f_2 & \longrightarrow & g \end{array} \right\}$$

$$S\text{-}B = \left\{ \begin{array}{lcl} Sp_{iii} & \longrightarrow & Sp_{ii} \\ S_1 & \longrightarrow & S_2 \\ f_1 & \longrightarrow & g \end{array} \right\}$$

Fig 21(d)
Example of diagram (cont.)

(c)

$$
\begin{tikzcd}
 & Sp_{iii} \arrow[d, "S-B"] \arrow[ddr, bend left, "C_{iii}"] & \\
Sp_i \arrow[r, "S-A"] \arrow[drr, "C_i"'] & Sp_{ii} \arrow[dr, "C_{ii}"] & \\
 & & Sp_{i+ii+iii}
\end{tikzcd}
$$

$C_i = \left(\begin{array}{l} Sp_i \longrightarrow Sp_{i+ii+iii} \\ s_1 \longrightarrow s_2 \\ f_1 \longrightarrow g \\ f_2 \longrightarrow g \end{array} \right)$

$C_{ii} = \left(\quad id \quad \right)$

$C_{iii} = S-B$

colimit

$Sp_{i+ii+iii} = \left(\begin{array}{l} \text{spec } Sp_{i+ii+iii} \text{ is} \\ \text{sort } \{ \text{sort } Sp_i . S_1, \ \text{sort } Sp_{iii} . S, \ \text{sort } S_2 \} \quad S_2 \\ \text{op } g : S_2 \to \text{boolean} \\ \text{axiom } g\text{-}g \text{ is} \\ \quad g(x) \Rightarrow g(x) \end{array} \right)$

Fig 21(e) Example of Diagram Extension (cont)

$$Sp_i + Sp_{iii} = \left\{ \begin{array}{l} \text{sort } Sp_i . S_1 \\ \text{sort } Sp_{iii} . S_1 \\ \text{op } Sp_i . f_1 : Sp_i . S_1 \to \text{boolean} \\ \text{op } Sp_{iii} . f_1 : Sp_{iii} . S_1 \to \text{boolean} \\ \text{op } Sp_i . f_2 : Sp_i . S_1 \to \text{boolean} \end{array} \right.$$

$$\Big\downarrow \; b_{i+ii+iii}$$

$$Sp_{i+ii+iii} = \left\{ \quad \text{See } 21(\text{?}) \quad \right\}$$

$$b_{i+ii+iii} = \left\{ \begin{array}{l} Sp_i . S_1 \longrightarrow S_2 \\ Sp_{iii} . S_1 \longrightarrow S_2 \\ Sp_i . f_1 \longrightarrow f \\ Sp_{iii} . f_1 \longrightarrow g \\ Sp_i . f_2 \longrightarrow g \end{array} \right\}$$

Fig 21(F) Example of Diagram Extension (cont)

(5)

After finishing the extension for each diagrams, let us use the following example:

Original diagrams:

$$\begin{tikzcd}
\boxed{d1: D1 \Longrightarrow SPEC} \arrow[r, "d12"] \arrow[d, "d13"] & \boxed{d2: D2 \Longrightarrow SPEC} \\
\boxed{d3: D3 \Longrightarrow SPEC} &
\end{tikzcd}$$

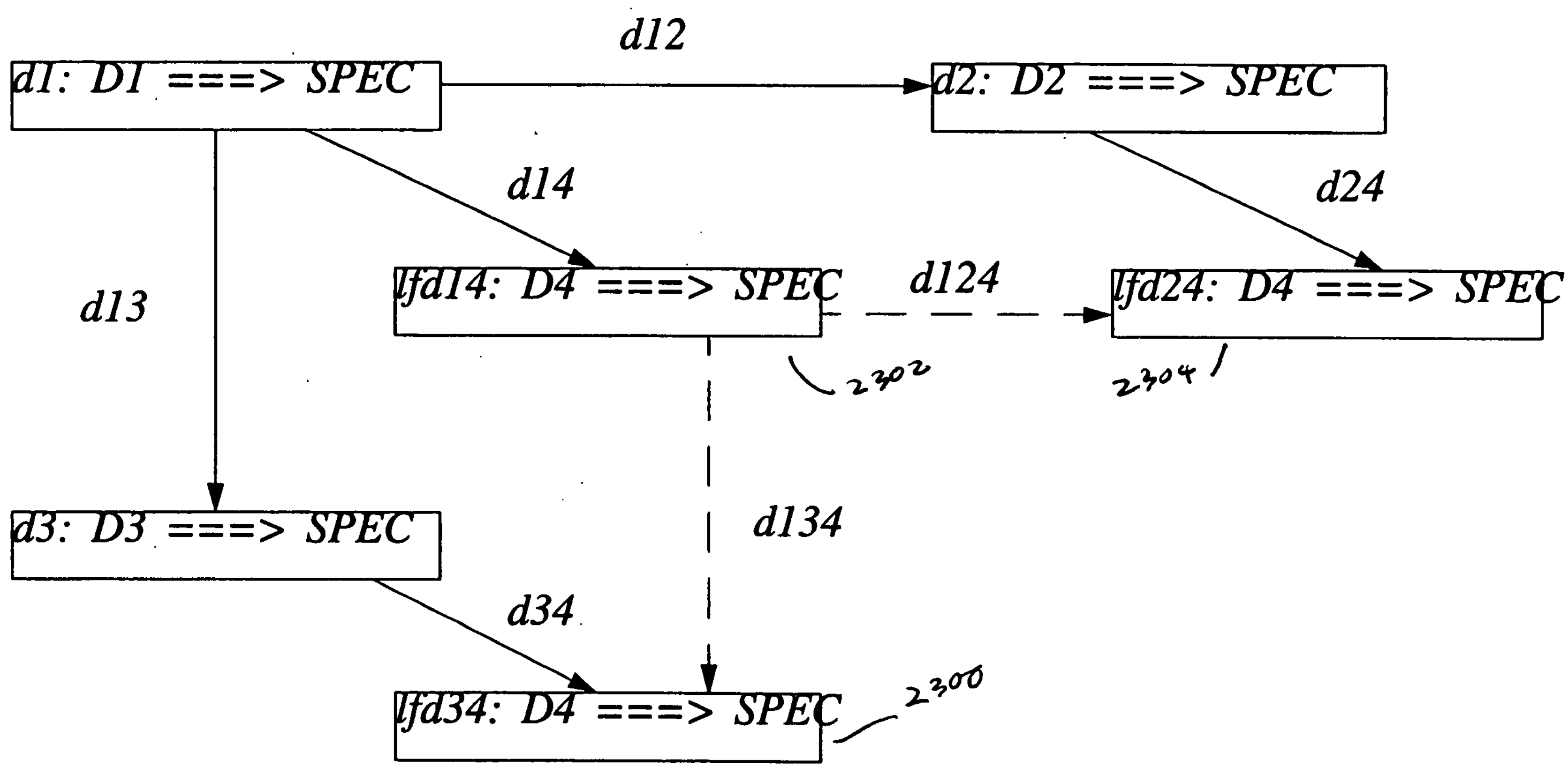
Its underlying shape categories, shape functors and the colimit are:

$$\begin{tikzcd}
\boxed{D1} \arrow[r, "F12"] \arrow[d, "F13"] \arrow[dr, "F14"] & \boxed{D2} \arrow[d, "F24"] \\
\boxed{D3} \arrow[r, "F34"] & \boxed{D4}
\end{tikzcd}$$

Part III

Fig. 22

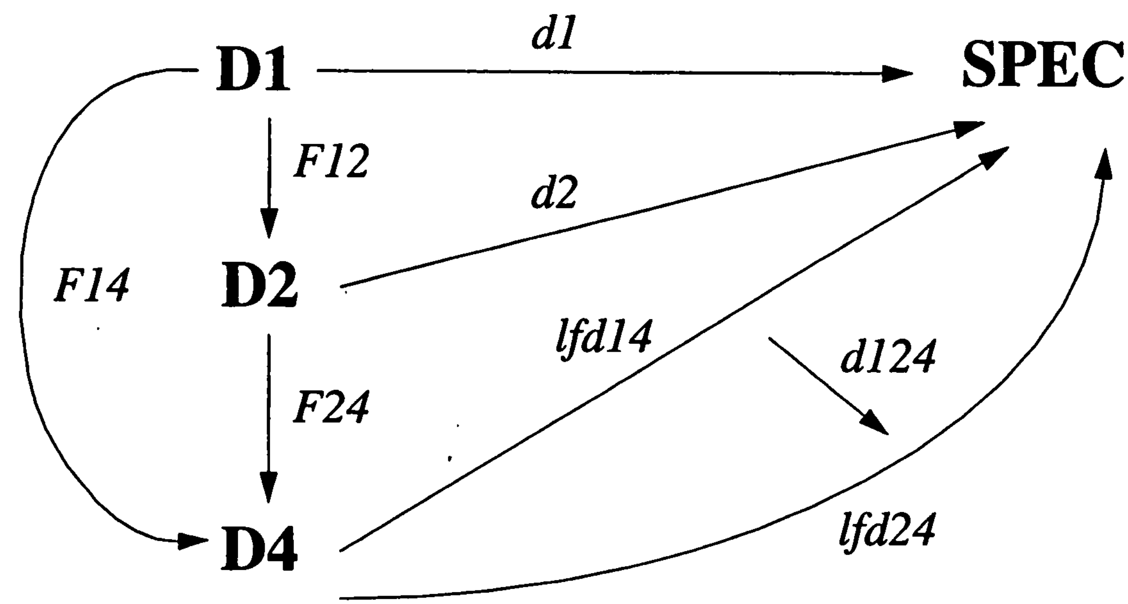
Extended diagrams:



The last algorithm step we are missing for constructing the diagram colimits is the diagram morphisms between extended diagrams. For example, the diagram morphism d124 and d134 (dotted lined arrows in above figure) are the ones needed.

Suppose lfd14 and lfd24 are two extensions of d1 and d2, given the colimit of the shape categories as D4. We would have the following picture.

Fig, 23



A morphism between lfd14 and lfd24 is a natural transformation which maps each node of D4 to an arrow in SPEC. We do th y the universal construction of witness arrows.

For any node ni in D4, we have F14(ni) = F12 o F24(ni). Let Sp1ni and Sp2ni two shape categories used for constructing a mapping for ni in its extension of d1 and d2, respectively, then we can have a shape function between Sp1ni and Sp2ni (inclusion, basically). That induces a diagram morphism between the base diagrams for the target of ni in lfd14 and lfd24, respectively. By imposing that diagram morphism and cocone morphism, we can get an unique arrow between lfd14(ni) and lfd24(ni). Repeating this process we construct a natural transformation between lfd14 and lfd24. Similarly, we can do this for any two extended diagrams.

The following flowchart is the algorithm for constructing a diagram morphism between two extended diagrams.

Fig. 24

09655179.091900

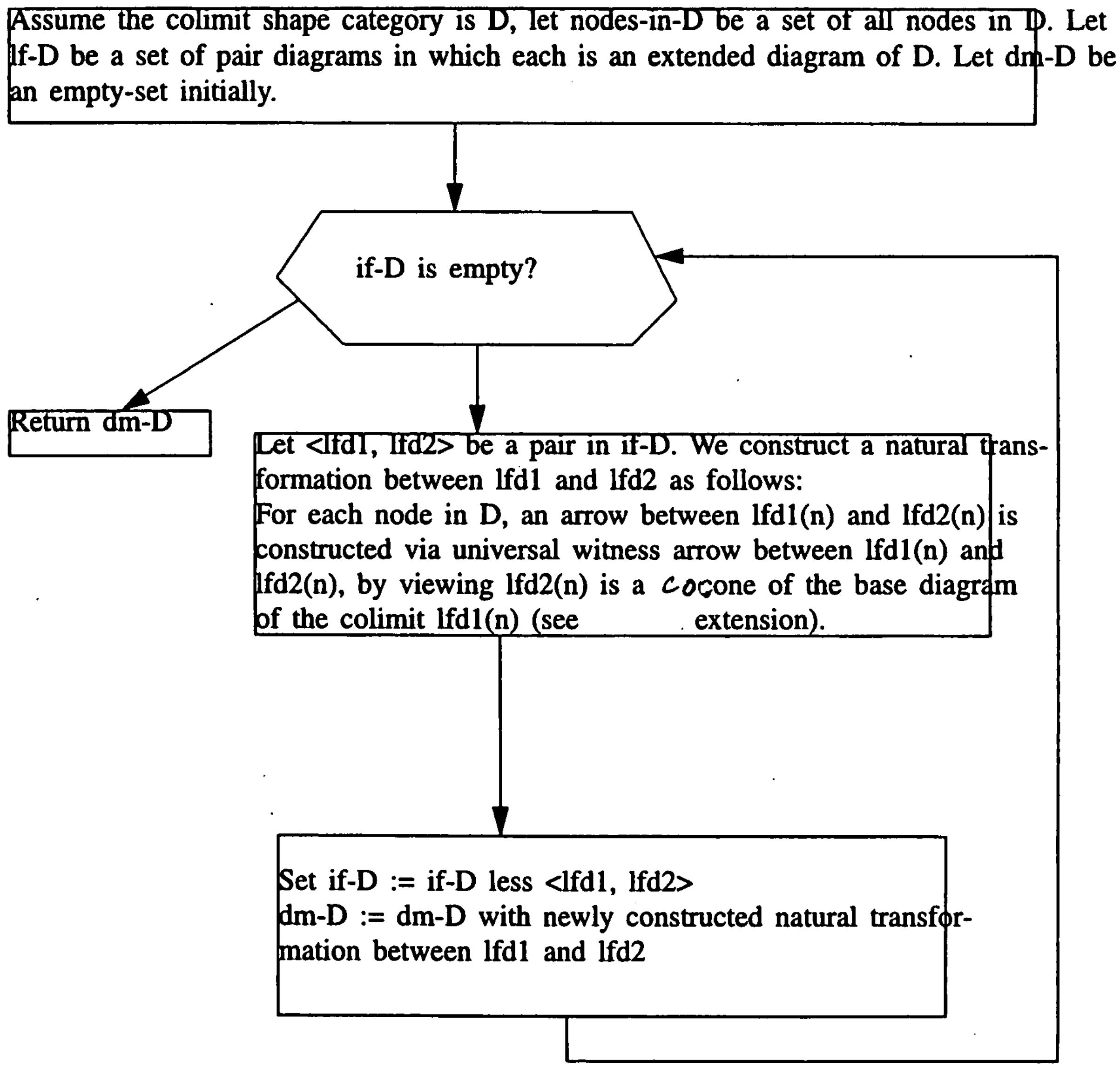

Assume the colimit shape category is D, let nodes-in-D be a set of all nodes in D. Let if-D be a set of pair diagrams in which each is an extended diagram of D. Let dm-D be an empty-set initially.
if-D is empty?
Return dm-D
Let <lfd1, lfd2> be a pair in if-D. We construct a natural transformation between lfd1 and lfd2 as follows:
For each node in D, an arrow between lfd1(n) and lfd2(n) is constructed via universal witness arrow between lfd1(n) and lfd2(n), by viewing lfd2(n) is a cocone of the base diagram of the colimit lfd1(n) (see extension).
Set if-D := if-D less <lfd1, lfd2>
dm-D := dm-D with newly constructed natural transformation between lfd1 and lfd2

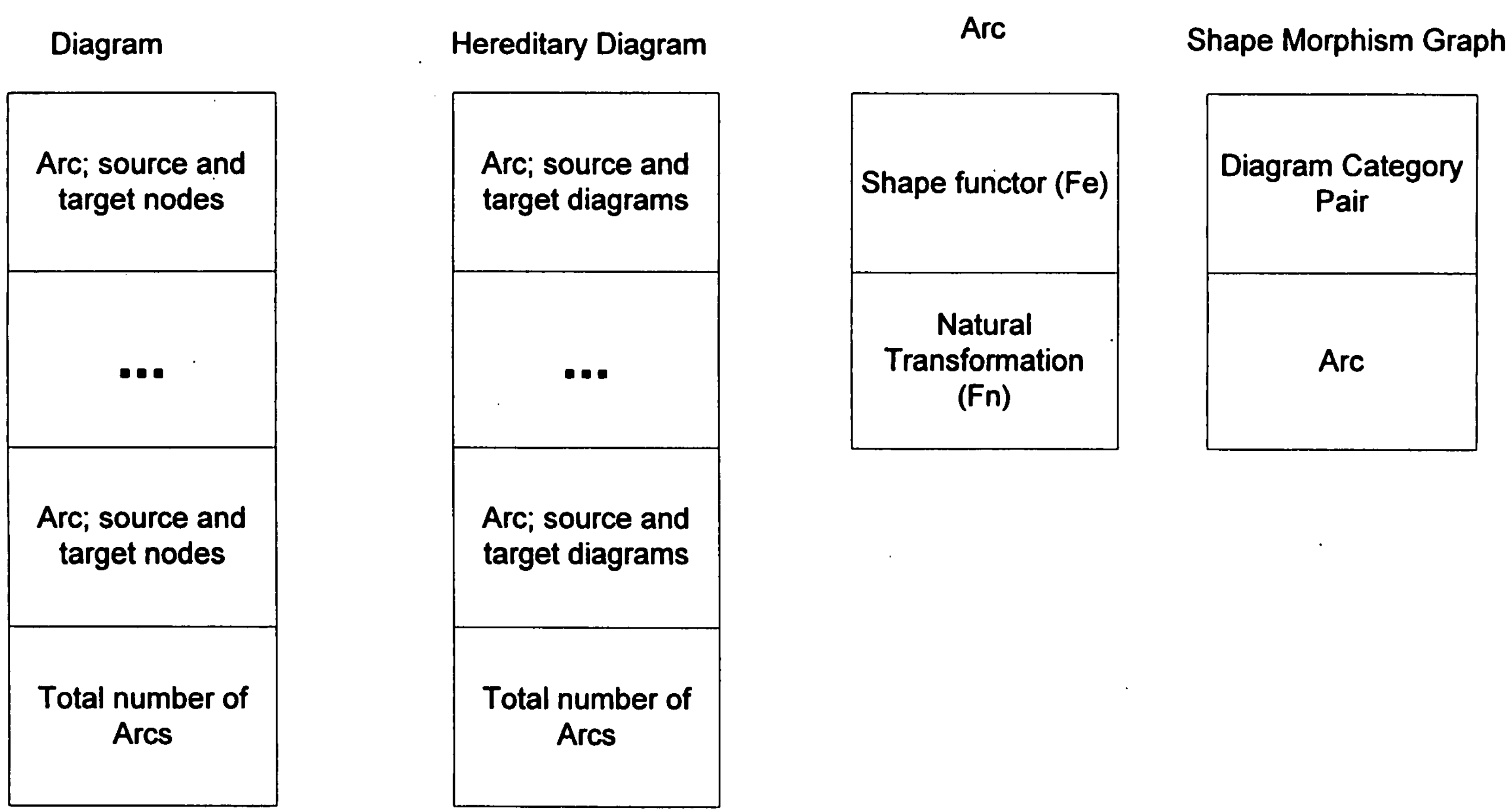
Fig. 25

The final step is to complete the colimit of the extended diagrams. The colimit is determined by computing the pointwise colimits over corresponding nodes in the extended diagrams. The morphisms are computed uniquely using universality of the pointwise colimits.

$$\begin{tikzcd}
\text{Extended diagram of d1} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \text{Extended diagram d2} \arrow[d, Rightarrow] \\
\text{Extended diagram d3} \arrow[r, Rightarrow] & \text{Colimit of Diagrams}
\end{tikzcd}$$

Taking Pointwise Colimit of Extended Diagrams (Can be done, since extended diagrams are all the same shape)

Fig. 26



Examples of Data Structures used in the Example Implementation
Fig. 27

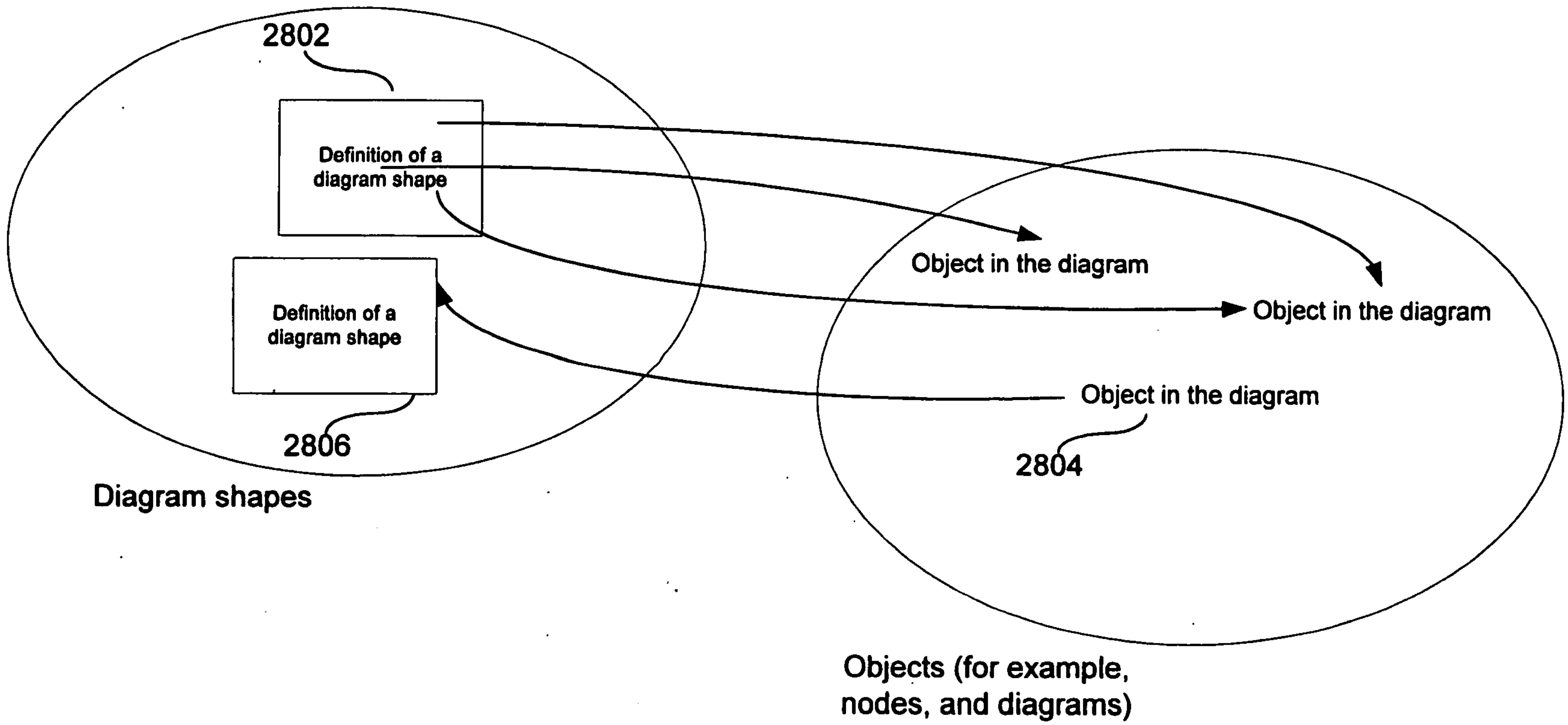

2802
Definition of a
diagram shape
Definition of a
diagram shape
2806
Diagram shapes
Object in the diagram
Object in the diagram
Object in the diagram
2804
Objects (for example,
nodes, and diagrams)

Fig. 28

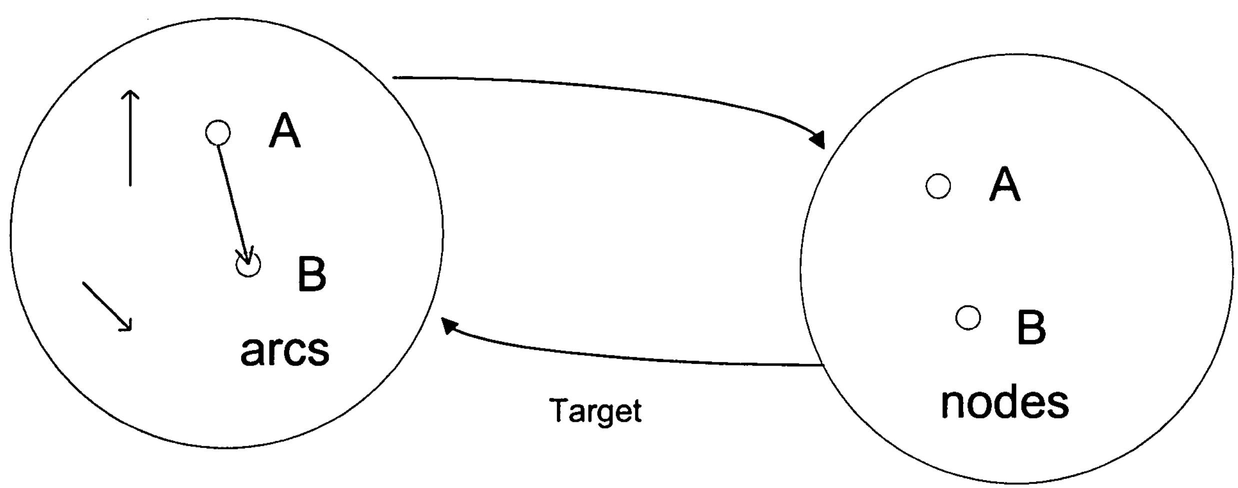


Fig. 29